**WAZIUP:** Co-funded by the Horizon 2020 Framework Programme of the European Union

# Open Innovation Platform for IoT-Big data in Sub-Sahara Africa

Grant Agreement Nº 687607

# Report D1.1

*Deliverable Title: Preparation and planning of the WAZIUP platform that responded to user's needs*

| | |
|---|---|
| Responsible Editor: | Part 1: C4A (Ousmane Samba)<br>Part 2: CREATE-NET (Corentin Dupont) |
| Contributors: | All Partners |
| Document Reference: | WAZIUP D1.1 – *Preparation and planning of the WAZIUP platform that responded to user's needs* |
| Distribution: | Public |
| Version: | 1.0 |
| Due Date: | M5 |

# EXECUTIVE SUMMARY

Most of ICT success stories in Africa address very concrete issues of local populations. For instance, it is reported that 70% of the population of Senegal relies on raising cattles as their main source of revenues. When those animals are stolen, some families are left in such dramatic situation that cases of suicide are not unheard of. The project DARAL was a first attempt to fight against cattle rustling with the help of technology. For instance, it provides a web application for cattle identification and is currently implemented in 5 zones with 1500 farmers and 18000 cattle registered. Agriculture is another very important source of activity and revenue. In most cases, farmers in Sub-Saharan Africa are lacking data on culture status. For instance, basic parameters, such as soil humidity and temperature, and sometimes more advanced parameters, such potassium and nitrogen levels, are very useful for increasing agriculture productivity. Farmers are also lacking actionable information on the condition of the farm. This actionable information can be displayed in the form of alerts, forecasts and recommendations. An example of such a service is a recommendation on the water levels needed for irrigation, taking into accounts the weather forecasts. Here, in addition to IoT technologies, Big Data technologies for data analysis could be integrated into the monitoring and production loop to provide predictions and feedbacks.

However, when discussing about massive deployment of IoT and Big Data, Africa's countries present problems such as lack of infrastructure, high cost of hardware, complexity in deployment and finally lack of technological eco-system. In Sub-Saharan Africa about 64% of the population is living outside cities. The region will be predominantly rural for at least another generation. The pace of urbanization here is slower compared to other continents, and the rural population is expected to grow until 2045. The majority of rural residents manage on less than few Euros per day. Rural development is particularly imperative where half of the rural people are depending on the agriculture/micro and small farm business, other half faces rare formal employment and pervasive unemployment. For rural development, technologies have to support several key application sectors like health, water quality, agriculture, livestock farming, climate changes, etc.

In order to bridge the gap between rural Africa need and the current IoT offer, the WAZIUP project aims at developing an IoT/Big Data platform that corresponds to users needs. This deliverable presents the first results of the Work Package 1 of the WAZIUP project. It includes the users and stakeholder requirements analysis, divided in several use case scenarios. Based on the user's requirements the technical specification of WAZIUP architecture is defined. The deliverable is divided into two parts:

- ***Part I: Definition of application cases and user's requirements, page 4***
- ***Part 2: Architecture requirements and specification, page 75***

Part 1 of the deliverable seeks to provide a comprehensive analysis of the uses cases. The use cases are collected through different channel: users/stakeholders interviews and survey, online literature and crowdsourcing. A collection of 31 use cases distributed in eight domains have been collected, described, discussed and analysed. They combine conceptual and functional descriptions on how the end user will interact with the system in order to complete the use case. Then using the WAZIUP requirement matrix and end user interview data, a selection matrix has been created in order to choose the most valuable use case that will be implement in our test bed use cases.

The real-life pilots are then described, aiming at validating the selected use cases and also validating the software and hardware platforms of WAZIUP. Each pilot is deployed in the premises/terrains of a supporting partner/stakeholder.

The part 2 describes the full architecture of the platform that will be developed. Based on the use cases described in part 1, we propose three concepts able to fulfil the requirements of end users: the Platform as a Service (PaaS) approach to IoT, the data processing capacity inspired from Big Data techniques and finally the local and global Cloud. The idea of extending the PaaS approach to IoT is to propose a platform dedicated to IoT developers that can reduce the time-to-market for an application by cutting the development costs. The Big Data techniques enable the processing of the huge amount of data produced by sensors. Those techniques allow creating actionable information and knowledge out of the raw data. Finally the local and global Clouds address the intermittent connection challenge: when Internet is not available, the user can still access some IoT functionalities from the local Cloud.

The architecture of the WAZIUP platform proposed in part 2 consists of a functional overview, the actors definitions, the components, the sequence diagrams, the detailed requirements for each of the components of the platform and the interfaces between them. The hardware platforms and networks are described, together with the data models and formats used by the platform. We then proceed to design each of the components. This part also gives a state of the art, both on IoT and Big Data.

# PART I: Definition of application cases and user's requirements

# TABLE OF CONTENTS

## LIST OF PICTURES



## LIST OF TABLES

# 1 INTRODUCTION

The goal of WP1, led by C4A, is to ensure that the solutions developed and deployed as part of WAZIUP directly and effectively respond to the needs and innovation opportunities of the rural end users. To accomplish this, and in order to maximize the potential for WAZIUP's wider impact, several case studies from a range of contexts across sub-Saharan Africa are assessed. A parallel goal is to ensure that the other WPs remain consistent in relation to WAZIUP's overall parameters, including the general architecture, definitions, assumptions, requirements, guidelines and common terminology to be adopted in the project.

The methodology used is a combination of Agile and Waterfall methods to firstly identify user stories which are part of an Agile approach. All user stories include a written sentence or two and, more importantly, a series of conversations about the desired functionality. Then it identifies the use cases, which are end-to-end sequence of interactions between an actor, and a system that yields a result of observable value to an actor, usually the instigating actor from the user story. The use cases are evaluated using a survey disseminated through our network of stakeholders. The most relevant use cases will be validated through 5 real-life pilots.

# 2 METHODOLOGY

Our methodology to identify and select the relevant use cases combine Agile and Waterfall design approaches:

- User stories are first identified. An agile approach is used to help shifting the focus from writing about requirements to talking about them. All agile user stories include a written sentence or two and, more importantly, a series of conversations about the desired functionality.
- Then uses cases are expressed into end-to-end sequence of interactions between an actor, and a system yielding to a value observable by an actor, usually the instigating actor from the user story.

A field analysis then allows selecting use cases having the highest acceptance within local communities. This is intended to support the business exploitation going to be driven through the Minimum Viable Product approach within the project.

## 2.1 User stories collection

The WAZIUP methodology of collecting user stories follows three channels: users/stakeholders interviews and survey, online literature and crowdsourcing.

- **Users/ stakeholders' interviews and survey**: the user stories collected with direct (face to face) interview, phone call, online survey with the users and stakeholder. WAZIUP African partners organized stakeholder physical meetings and phone calls. Farmerline created a mobile application to collect the users' needs. This online survey tool based on mobile is used to collect the feedback. This online survey is done under task 5.1. The interview and survey results can be found in the Section 4.
- **Online literature:** With this channel the user stories are collected from online literatures, books, articles and previous projects. The WAZIUP project performed a review of online literatures for state-of-the art IoT big data use case in African context (see Section 9 in part 2).
- **Crowdsourcing:** The project organized two public events: one in Senegal and another in Ghana, in order to collect the user stories. The launch event of the WAZIUP program was organized at the Novotel hotel in Dakar on February 4th 2016. More than 100 participants including members of government, private sector, EU delegation in Dakar and tech/start-up community attended it. It allowed Abdur Rahim, coordinator, to present the project, its goals, objectives and methodologies. It also allowed the public to be aware of the existence of that project and of the partners behind its realization. The second event was organized at iSpace in Accra, Ghana on June 17th 2016. More than 30 participants coming mostly from Ghana developer community, allowed project coordinator Abdur Rahim, to present the project, its goals, objectives and methodologies. A very interesting question and answers session from the public was performed on the technologies used.

## 2.2 Use case description

The collected stories are described in two levels: i) User Story and ii) Use Cases. The Figure 1 illustrates these two levels. The user story is the conceptual level and use case is the functional level.

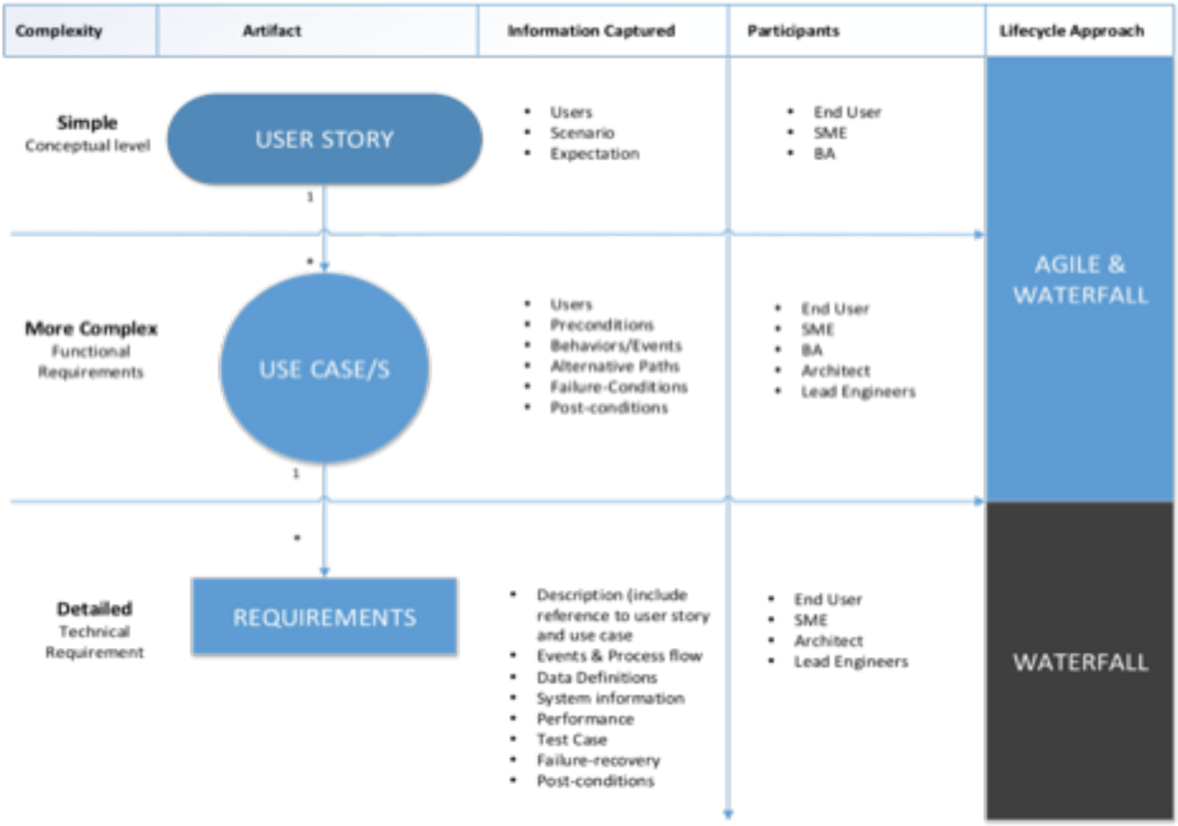


**Figure 1: Methodologies**

The User Stories are described in the conceptual level. In this level, we identify the users and their expectation. Here users are end-users, SME, BA (Business Analyst). User stories are part of an Agile approach that helps shift the focus from writing about requirements to talking about them. All Agile user stories include a written sentence or two and, more importantly, a series of conversations about the desired functionality.

The Use Cases describes the functional requirements of the user stories. Use cases are the end-to-end sequence of interactions between an actor, and a system that yields a result of observable value to an actor, usually the instigating actor from the user story. It describes the users condition, behaviour/events and actions.

## 2.3 Analysis of user stories

Once the use cases have been identified and described, the following table matrix is used to analyse them. This analysis includes the technical requirement of WAZIUP platform, IoT and Big Data, and also business potential of the use case.

**Table 1: Meta-data related to business potential of use case**

| ID | Meta-data title | Description |
|---|---|---|
| **B1** | End-user impact | How will end-users experience this use case? |

| | | |
|---|---|---|
| **B2** | Service impact | How will service provider experience this use case? |
| **B3** | Overall impact | How important is the problem (Low, Moderate, High) |
| **B4** | Privacy concern | Is there any privacy concern related to this use case? |
| **B5** | Actors | Who are the main actors? |

**Table 2: Meta-data related to WAZIUP platform**

| ID | Meta-data title | Description |
|---|---|---|
| **T1** | Data source | What kind of data sources are required to realize this use case? |
| **T2** | User interaction | How users will interact with the application? |
| **T3** | IoT sensor | Does this use case relate to IoT? |
| **T4** | Data analysis | What kind of big data problems are associated with this use case? |
| **T5** | Local and global automation | Does this application need local or global automation or both? |
| **T6** | Infrastructure | What kind of infrastructure is required to realize this application? Can it be implemented with existing infrastructure? |

The requirement matrix and end user interviews will be used to create a selection matrix in order to choose the most valuable use case that will be implement in our test bed use cases. Section 5 describes the selected use case that will be implemented in this project.

# 3 USE CASES

In this section, the detailed descriptions of all the application domains and use cases are provided. There are eight domains: agriculture, fish farming, water, health, environment, logistics and transports, Urban and domestic and finally cattle rustling.

## 3.1 Agriculture

This section describes the use case analysis of the agriculture application domain.

### 3.1.1 Problem context

The Agriculture domain depends on numerous factors, such as soil quality (such as. water retention capacity, organic matter content), weather (intra-seasonal and inter-annual climatic variability) and agro-management decisions (such as date of sowing, irrigation, fertilizing, tilling).

The farms in Africa require more automation to reduce losses and boost productivity. Precision farming is the ability to handle variations in productivity, maximize financial return, reduce waste and minimize impact on the environment; using data, sensing and communication technology. Thus, using data analysis to customize as well as improve overall operations of farming. The proposed activity may include monitoring of soil moisture, continuous monitoring of water storage tanks and field conditions (weather/rainfall, soil moisture, electrical conductivity, etc.). For example, farmers will be able to monitor the temperature of grain bins (storage areas) and receive an alert if the temperature rises outside of an acceptable range. In turn, this will help preserve grains in storage areas. The same approach can be extended to alerts for pest control operations. An emerging class of entrepreneur, the so-called agropreneur, is rising to the challenge and providing value to small scale farmers by offering technical assistance and even taking responsibility for farm productivity during the growing season.

Using wireless sensor networks in Precision Farming applications will revolutionize the data collection in agricultural field. Highly automated agriculture system requires intensive sensing of environmental conditions at the ground level and rapid communication of the raw data. Depending on the availability of computational and storage power, the data is sent to a local or remote server. The identification of threats for the crops, the decision making, and the control of farm equipment is done in real time. For example, automated actuation devices like sprinklers, foggers, valve-controlled irrigation system can be used to control irrigation, fertilization and pest levels.

### 3.1.2 User stories and use cases

Table 3 describes the user stories of the agriculture domain.

**Table 3: Agriculture user stories**

| ID | User story | Actors |
|---|---|---|
| **UC1** | I want to know the various soil situation on my farm so that I don't waste inputs and water | Farmer |

| | | |
|---|---|---|
| **UC2** | As service provider, I want to know the precise weather details to provide accurate advisory service to Farmers | Weather information Service Provider / Farmer? |
| **UC3** | I want to be able to measure the temperature and moisture of a storage area or commodity. | Farmer/ Aggregators |

#### *3.1.2.1 Use Case 1: Monitor soil*

In the following table, the detailed description of the use case 1 is outlined.

**Table 4: Agriculture use case 1**

| **Name** | **Monitor soil** |
|---|---|
| **Description** | The farmer will get notification about his soil situation and information's about possible actions to take. I want to know the the soil composition on my farm so that I do not waste inputs and water (within 2 acres - soil moisture and nutrients: potassium, nitrogen). |
| **Users** | Farmer |
| **Preconditions** | Farmer has sensors installed on his farm. |
| **Basic Course of Events** | 1. The farmer requests notification to WAZIUP through SMS /USSD<br>2. The system requests identity verification<br>3. The user sends credential.<br>4. Soil data and possible actions against it are sent to the user |
| **Alternative Paths** | In step 2, if the authentication fails the system send to the user necessary information to register.<br>In step 4, If the soil data are not available, the system sends notification asking user to try again later. |
| **Post conditions** | User is updated with most recent soil data, and recommendations based on Big Data analysis. |

#### *3.1.2.2 Use case 2: Monitor Field Weather Data*

In the following table, the detailed description of the use case 2 is outlined.

**Table 5: Agriculture use case 2**

| **Name** | **Monitor Field Weather Data** |
|---|---|
| **Description** | The service provider will receive precise weather details from sensors. |
| **Users** | Weather Service provider, Farmer |

| | |
|---|---|
| **Preconditions** | Weather service provider have sensors network providing different weather measure.<br>The service provider has access to WAZIUP Web API. |
| **Basic Course of Events** | 1. The service provider request recent weather detail about the targeted area to WAZIUP through web API.<br>2. The system verify the user identity<br>3. Most recent weather details are sent to the weather service provider in a readable format. |
| **Alternative Paths** | In step 2, if the authentication fails the system send to the user necessary information to register.<br>In step 3, If the weather data are not available, the system send notification about the data status. |
| **Post conditions** | Service provider receives accurate weather data for the targeted area. |

#### *3.1.2.3 Use case 3: Storage Moisture and Temperature*

In the following table, the detailed description of the use case 3 is outlined.

**Table 6: Agriculture use case 3**

| **Name** | **Storage Moisture and Temperature** |
|---|---|
| **Description** | The farmer will get notification about his storage area/commodity moisture and temperature and about possible actions to take.<br>The farmer wants to be able to know or measure the temperature and moisture of a storage area or commodity so that I can prevent crops from going bad. |
| **Users** | Farmer |
| **Preconditions** | Farmer has moisture and temperature sensors installed in his storage area and precondition to send a message when it get to a certain maximum level. |
| **Basic Course of Events** | 1. Farmer must already be registered with a pre-conditioned moisture and temperature levels.<br>2. The system sends a notification via SMS/calls when the moisture and temperature are at a maximum level.<br>3. Storage area conditions and possible actions are sent to the user |
| **Alternative Paths** | 1. The farmer request notification to WAZIUP through SMS / USSD<br>2. In step 2, if the system is unable to send the notification to the user about the storage conditions.<br>3. In step 1, If the storage conditions are not available, the system send notification asking user to try again later. |
| **Post conditions** | User is updated with current conditions of his storage area and commodity, |

| | |
|---|---|
| | and possible actions |

### 3.1.3 Use case diagrams and actor description

Three main use cases have been identified for the Agriculture domain. Two main actors Farmers and Service providers operate these use cases.

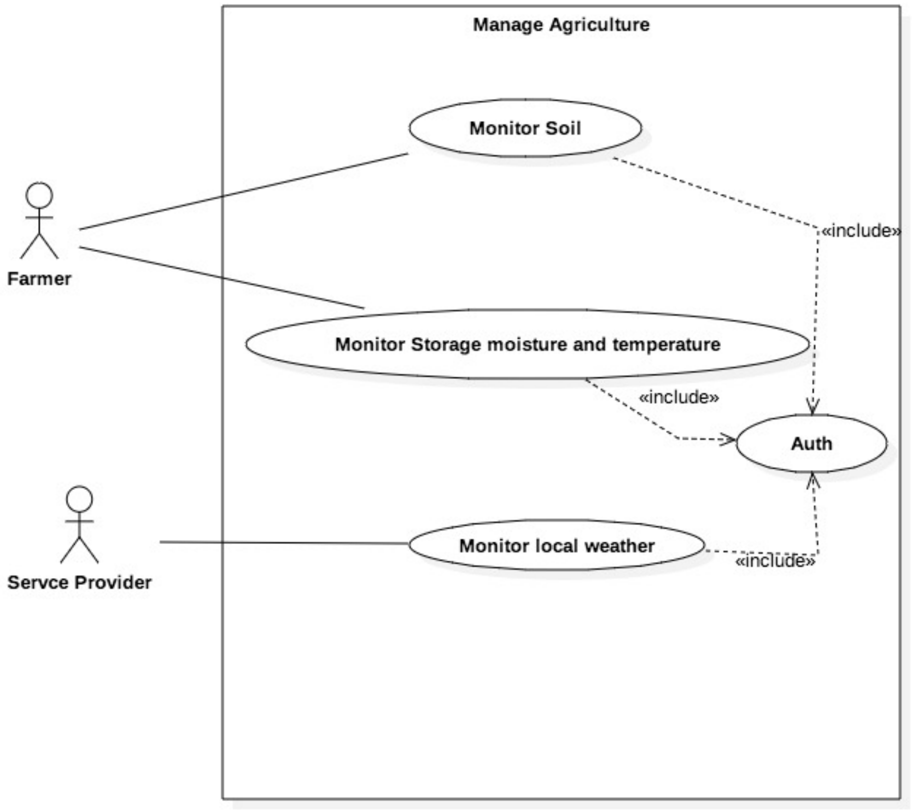


**Figure 2 : Agriculture use case actor diagram**

### 3.1.4 Use case analysis

In the following tables, each of the use case is analysed with its potential for the business cases and also how these use cases are mapped to WAZIUP technical requirements.

**Table 7: Agriculture use case analysis**

| **Business Related Meta-Data** | | | | | |
|---|---|---|---|---|---|
| **Uses Cases** | **B1 (end user impact)** | **B2 (service provider impact)** | **B3 (overall impact)** | **B4 (privacy)** | **B5 (actor)** |
| **UC1 Monitor soil** | Monitoring soil moisture and nutrients will help farmer be more | Service provider can also benefit from farmers soil data (big | Soil quality is the ideal indicator of sustainable land | No privacy concerns of the farmers | Farmer |

| | | | | | |
|---|---|---|---|---|---|
| | productive and self efficient in inputs use. | data) Ex: future need of fertilizer | management (HIGH) | | |
| **UC2 Monitor Field Weather Data** | Through local weather data, Farmer will be able to make better decision in crop and water management. | Service provider can also benefit from farmers local weather station (big data), to give better farmer specific weather prediction | Access to seasonal weather forecasts is helping small-scale farmers make informed cropping decisions (HIGH) | No privacy concerns of the farmers | Farmer, Service Provider |
| **UC3 Storage Moisture and Temperature** | Through moisture and temperature sensors, farmer will be able to better monitor and handle seed and grain storage to avoid issues. | Service provider can also benefit from storages facilities data (Big Data), to better predict food quantities. | Better storage monitoring can reduce post-harvest food loss. (HIGH) | No privacy concerns of the farmers | Farmer, Service Provider |

| **Technical Meta-Data** | | | | | | |
|---|---|---|---|---|---|---|
| **Uses Cases** | **T1 (Data Source)** | **T2 (User interaction)** | **T3 (IoT sensor)** | **T4 (Data analytic)** | **T5 (local and global)** | **T6 (infrastructure)** |
| **UC1 Monitor soil** | This use case requires different sensors providing different soil measurement (soil moisture and nutrients: potassium, nitrogen) | Farmer: SMS / USSD, Service provider: Web/Mobile web | Yes this Use case is related to IoT | This use case can use Big data to allow prediction of inputs procurement. | Local | Sensors, gateway + SMS based app / Web/Mobile App |
| **UC2 Monitor Field Weather Data** | This use case require different sensors providing local weather forecasting | Farmer: SMS / USSD, Service provider: Web/Mobile web | Yes this Use case is related to IoT | This use case can use Big data to allow prediction of inputs procurement. | Local | Sensors, gateway + SMS based app / Web/Mobile App |
| **UC3 Storage Moisture and Temperature** | This use case require different sensors providing temperature | Farmer: SMS / USSD, Service provider: Web/Mobile web | Yes this Use case is related to IoT | This use case can use Big Data to allow better prediction of food quantities and seasonal | Local | Sensors, gateway + SMS based app / Web/Mobile App |

| | | | | | | |
|---|---|---|---|---|---|---|
| | and moisture data | | | agriculture productivity estimation. | | |

## 3.2 Fish farming

This section describes the use case and users stories, use case analysis of the Fish Farming application domain.

### 3.2.1 Problem context

Internet of Things has already expanded into many parts of the agricultural world, from wine vineyards to dairies. It's only natural that it expands into aquaculture. African countries could reconcile the soaring demand for fish with the need to safeguard biodiversity and make the fishing sector sustainable by harnessing a raft of new technologies, according to a report by the Food and Agriculture Organization (FAO).

The aim is to use a network of sensors to monitor the fish ponds in real time, measuring water temperature, quality, oxygen levels and other parameters. This will improve the current farm management process by giving farmers the ability to monitor their ponds remotely and in a near real-time way.

### 3.2.2 User stories and use cases

The details description of the user stories is outlined table below.

**Table 8: Fish Farming user stories**

| ID | User story | Actors |
|---|---|---|
| **UC1** | As fish farmer I want to know about water quality, dissolved oxygen, water temperature) in my fish pond and the information about possible actions to take. | Fish Farmer |
| **UC2** | As fish farmer I want to know the number of fish in my fish pond. | Fish Farmer |
| **UC3** | As fish farmer I want a cost efficient feeding system. | Fish Farmer |
| **UC4** | As fish farmer I want a Plankton measurement system so that i can reduce the fish feeding cost. | Fish Farmer |

#### *3.2.2.1 Use case 1: Fish Pond Water Quality*

In the following table, the detailed description of the use case 1 is outlined.

Table 9: Fish Farming use case 1

| | |
|---|---|
| **Name** | **Fish Pond Water Quality** |
| **Descriptions** | I want to get notification about the water quality in my fish pond and information about possible actions to take. As fish farmer I want to know about water quality, dissolved oxygen, water temperature. The farmer can control the water level, the amount of feeding they do and whether they need to treat the water "if contaminated". Thus help them improve their productivity by reducing the rate of which their fish is dying. |
| **Users** | Fish Farmer |
| **Preconditions** | Fish farmer have sensors installed on his fish pond. |
| **Basic Course of Events** | 1. The farmer request notification to WAZIUP through SMS /USSD<br>2. The system request identity verification<br>3. The user sends credential.<br>4. Fish pond water quality data and possible actions against it are sent to the user |
| **Alternative Paths** | In step 2, if the authentication fails the system send to the user necessary information to register.<br>In step 4, If the water quality data are not available, the system send notification asking user to try again later. |
| **Post conditions** | User is updated with most recent fish pond water quality data, and possible actions |

#### *3.2.2.2 Use case 2: Estimate fish quantity*

In the following table, the details description of the user story 2 is outlined.

Table 10: Fish Farming use case 2

| | |
|---|---|
| **Name** | **Estimate fish quantity** |
| **Descriptions** | I want to be able to know in real time an estimated quantification of the fish in my pond. As a farmer, knowing the estimate number of fish in a pond can be helpful in order to control quantity of food that are distributed to each fish pond. |
| **Users** | Fish Farmer |
| **Preconditions** | Sensors network is installed on the fish pond and send data to WAZIUP platform |
| **Basic Course of Events** | 1. The farmer request notification to WAZIUP through SMS /USSD<br>2. The system request identity verification |

| | |
|---|---|
| | 3. The user sends credential.<br>4. Fish quantity data estimation is sent to the user |
| **Alternative Paths** | In step 2, if the authentication fails the system send to the user necessary information to register.<br>In step 4, If the water quality data are not available, the system send notification asking user to try again later. |
| **Post conditions** | The appropriate updated data are sent to the fish farmer |

#### *3.2.2.3 Use case 3: Cost-Efficient Feeding System*

In the following table, the details description of the user story 3 is outlined.

**Table 11: Fish farming use case 3**

| **Name** | **Cost-Efficient Feeding System** |
|---|---|
| **Descriptions** | As fish farmer, I want a cost efficient feeding system which is most important for professional Fish Farming. Fish feeding typically accounts for 50% to 80% of a fish farm's overhead costs, overfeeding means much of the feed goes to waste, hurting the bottom line of the business. Automating the feeding system can help address these issue. |
| **Users** | Fish farmer |
| **Preconditions** | Farmer has an automated mechanism using sensor installed, along with plankton measurement and fish estimation. |
| **Basic Course of Events** | 1. Sensor data (plankton, fish behaviour, fish count) are analyzed to determine fish appetite<br>2. Once the status is ok for feeding, signal is sent to the feeding mechanism<br>3. The mechanism selects the right quantity and put it into the fish pond. |
| **Alternative Paths** | No |
| **Post conditions** | The fish receive the right quantity of food. |

#### *3.2.2.4 Use case 4: Plankton measurement for fish feeding supply accuracy.*

In the following table, the details description of the use case 4 is outlined.

**Table 12: Fish farming use case 4**

| **Name** | **Plankton measurement for fish feeding supply accuracy** |
|---|---|
| **Descriptions** | Plankton is a one of the main sources of food for fish. They are the most common prey for all fish larvae. Plankton has its place in the lower |

| | |
|---|---|
| | regions of the food chain and is the basic source of food for small aquatic animals like fish larvae. During the early stage of their life cycle fish rely on their yolk sac for nutrition. They also rely on plankton to survive during its development stage. And if the number of plankton decreases, the population of fishes will be greatly affected. This cycle clearly demonstrates the impact of plankton upon pond life. Fish farmers have increased fish yields in ponds by using inorganic or chemical fertilizers and organic fertilizers or "manures." |
| **Users** | Fish farmer |
| **Preconditions** | Fish farmer have sensors installed on his fish pond. |
| **Basic Course of Events** | 1. The farmer request notification to WAZIUP through SMS /USSD<br>2. The system request identity verification<br>3. The user sends credential<br>4. Fish pond water quality data and possible actions against it are sent to the user |
| **Alternative Paths** | In step 2, if the authentication fails the system send to the user necessary information to register.<br>In step 4, If the water quality data are not available, the system send notification asking user to try again later. |
| **Post conditions** | User is updated with most recent fish pond water plankton information data, and possible actions |

### 3.2.3 Use cases diagram and actor's description

Four use cases have been identified for fish farming with one actor: the fish farmer.

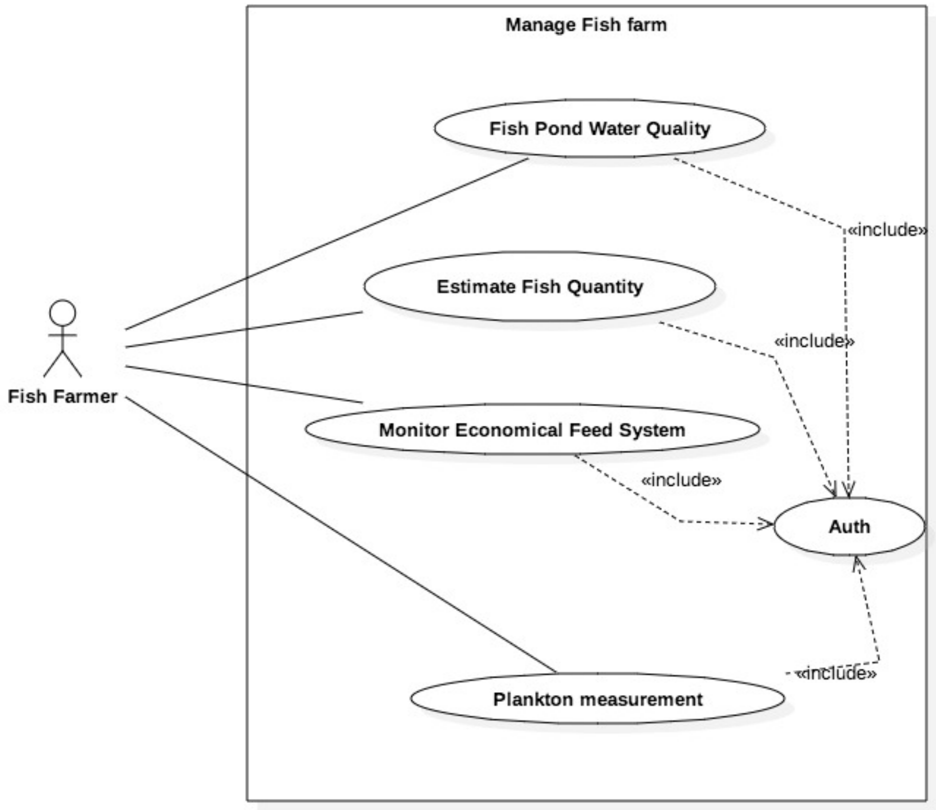


Figure 3: Fish farming use case actor diagram

### 3.2.4 Use cases analysis

In the following tables each of the use case is analysed with its potential for the business cases and also how the use case mapped to WAZIUP technical requirements.

Table 13: Fish farming use case analysis

| **Business Related Meta-Data** | | | | | |
|---|---|---|---|---|---|
| **Uses Cases** | **B1 (end user impact)** | **B2 (service provider impact)** | **B3 (overall impact)** | **B4 (privacy)** | **B5 (actor)** |
| **UC1 Fish Pond Water Quality** | Because fish are totally dependent upon water to breathe, feed and grow, excrete wastes, maintain a salt balance, and reproduce, understanding the physical and chemical qualities of water is critical to successful aquaculture. | Service provider can leverage big data information on large scale fish pond around an area to adapt their offer to fish farmer. | Water quality records will allow farmers to note changes and make decisions fast so that corrective actions can be taken quickly (HIGH). | No privacy concerns of the farmers | Fish farmer |

| | | | | | |
|---|---|---|---|---|---|
| **UC2 Estimate Fish Quantity** | Without reliable estimates of the numbers and size distributions of fish in the ponds, producers cannot accurately project harvests. | Service provider can apply the data analytic application to estimate the fish quantity and this information can be provided to farmer as a value added service. | For now, harvesting and completely draining a pond is the most reliable method for determining fish inventory, even though it exist other estimations method, having a sonar counting devices may provide a fast and accurate way to inventory fish pond (MEDIUM). | No privacy concerns of the farmers | Fish Farmer |
| **UC3 Cost-Efficient Feeding System** | Fish feeding typically accounts for 50% to 80% of a fish farm's overhead costs, but since feeding is a manual task, it's mostly an unmeasured and inexact method with one of two outcomes: overfeeding or underfeeding. | Service provider can apply big data tools and advanced algorithm to generate the knowledge to guide farmer on appropriate cost-effective feeding system. | An automatic smart feeder is designed to address these issues and dispense just the right amount of feed by using sensors that measure fish appetite. The impact will be great on costs control. (MEDIUM) | No privacy concerns of the farmers | Fish Farmer |
| **UC4 Plankton measurement for fish feeding supply accuracy** | Combined to previous Use Case | Combined to previous Use Case | MEDIUM | No privacy concerns of the farmers | Fish Farmer |

| **Technical Meta-Data** | | | | | | |
|---|---|---|---|---|---|---|
| **Uses Cases** | **T1 (Data Source)** | **T2 (User interaction )** | **T3 (IoT sensor)** | **T4 (Data analytic)** | **T5 (local and global)** | **T6 (infrastructur e)** |
| **UC1 Fish Pond Water Quality** | This use case require sensors providing different water quality measure (pH, water level, Turbidity, temperature, DO, salinity) | Farmer: SMS / USSD , Service provider: Web/Mobil e web | Yes this Use case is related to IoT | This use case can use Big data to allow better decision on the Fish farming supply chain. | local | Sensors, gateway + SMS based app / Web/Mobile App |
| **UC2 Estimate Fish Quantity** | This use case requires different | Farmer: SMS / USSD | Yes this Use case | This use case can use Big data | Local | Sensors, gateway + SMS |

| | | | | | | |
|---|---|---|---|---|---|---|
| | sensors to estimate fish count in a pond. | , Service provider: Web/Mobile web | is related to IoT | (advanced data analytic) to allow prediction of inputs procurement. | | based app / Web/Mobile App |
| **UC3 Cost-Efficient Feeding System** | This use case require different sensors providing information on fish behaviour to determine if they are hungry or not, and also plankton measurement. | Farmer: SMS / USSD , Service provider: Web/Mobile web | Yes this Use case is related to IoT | This use case can use advanced data analytic algorithm to allow prediction of inputs procurement. | Local | Sensors, gateway + SMS based app / Web/Mobile App |
| **UC4 Plankton measurement for fish feeding supply accuracy** | Combined to previous Use Case. | Farmer: SMS / USSD , Service provider: Web/Mobile web | Yes this Use case is related to IoT | This use case can use Big data to allow prediction of inputs procurement. | Local | Sensors, gateway + SMS based app / Web/Mobile App |

## 3.3 Water

This section describes the use case and users stories of the water application domain.

### 3.3.1 Problem context

As one of the most important natural resources, the management of water is becoming increasingly an important as water resources are growing scarcer. This is especially the case for rural areas and developing countries, such as countries in Africa. The proposed activities include monitoring community water pond quantity and quality. The water infrastructure based on IoT will help improve a good water quality, supply, treatment, transportation and storage. The priority for action should be to deploy the IoT sensors at the infrastructure level since the water savings will be the greatest and action should be the fastest. A utility should be able to justify the expenditure on the water savings particularly within the current context of scarcity. A water management system based on IoT for rural areas and developing countries can help the Community representative monitor and manage water use more efficiently.

### 3.3.2 User stories and use cases

In the following table, the user stories of the water are listed.

**Table 14: Water user stories**

| ID | User stories | Actors |
|---|---|---|
| **UC1** | As community rep I want to have the ability to know the | Community rep |

| | | |
|---|---|---|
| | amount of water in ponds/tanks. | |
| **UC2** | As community rep/farmer I want to know the quality of water in ponds/tanks | Farmer/Community rep |
| **UC3** | Prediction of the duration of water reserve | Community rep |

#### *3.3.2.1 Use case 1: Measure Water Level*

In the following table, the details description of the user story 1 is outlined.

**Table 15: Water use case 1**

| **Name** | **Measure Water Level** |
|---|---|
| **Descriptions** | I want to get notification about the amount of water in my pond and information about possible actions to take.<br>This use case offers the ability to know the amount of water per pond/tank and water consumption daily in order to predict availability.<br>For a water-supply system fulfils its public health function depends almost directly with the efficiency and effectiveness of its management. Measure water consumption in order to predict availability. |
| **Users** | Community representative |
| **Preconditions** | Sensors and gauges installed on water pond/tank. |
| **Basic Course of Events** | 1. User is registered and link to the particular tank/sensor<br>2. Daily notification (SMS/voice) on water level and consumption is sent to the user<br>3. Also, the use request notification to WAZIUP through SMS /USSD/web<br>4. The system request identity verification<br>5. The user send credential.<br>6. Pond/tank water level/consumption is sent to the user |
| **Alternative Paths** | In step 4, if the authentication fails the system send to the user necessary information to register.<br>In step 5, If the water level/consumption data is not available , the system send notification asking user to try again later. |
| **Post conditions** | User is updated with updated pond water level/consumption data |

#### *3.3.2.2 Use case 2: Water Quality*

In the following table, the detailed description of the use case 2 is outlined.

Table 16: Water use case 2

| Name | Water Quality |
|---|---|
| Descriptions | I want to know the water quality in a pond and information's about possible actions to take.<br>This use case offers the ability to know the quality of water in pond (drinkable water or not, acidity of water is good for irrigation or feeding the farm animals) |
| Users | Farmer /Community Representative |
| Preconditions | Sensors installed in water pond |
| Basic Course of Events | 1. The use send request notification to WAZIUP through SMS /USSD/web<br>2. The system request identity verification<br>3. The user send credential.<br>4. Pond water quality is sent to the user |
| Alternative Paths | In step 2, if the authentication fails the system send to the user necessary information to register.<br>In step 4, If the water quality data is not available, the system send notification asking user to try again later. |
| Post conditions | User is sent the most recent pond water quality data |

#### 3.3.2.3 Use case 3: Prediction of the duration of water reserve

In the following table, the details description of the use case 3 is outlined.

Table 17: Water use case 3

| Name | Prediction of the duration of water reserve |
|---|---|
| Descriptions | I want to get notification on how long a water reserve will last and the ability to predict how long the current water reserve will last for a community. To know the capacity of the water pond/tank and monitor the water consumption per day for a community. |
| Users | Community Representative |
| Preconditions | Sensors and gauges installed on water pond/tank |
| Basic Course of Events | 1. User is registered and link to the particular tank/sensor<br>2. Daily/weekly notification (SMS/voice) on how long the water reserve will last is sent to the user<br>3. Also, the use request notification to WAZIUP through SMS /USSD/web<br>4. The system request identity verification<br>5. The user send credential. |

| | |
|---|---|
| | 6. The prediction on the water quantity and is sent to the user |
| **Alternative Paths** | In step 4, if the authentication fails the system send to the user necessary information to register.<br>In step 5, If the water level/consumption data is not available, the system send notification asking user to try again later. |
| **Post conditions** | User is updated with latest prediction for the water pond/tank data |

### 3.3.3 Use case diagram and actor description

Three use cases have been identified for water, operated by two main actors: Farmer and Community representative.

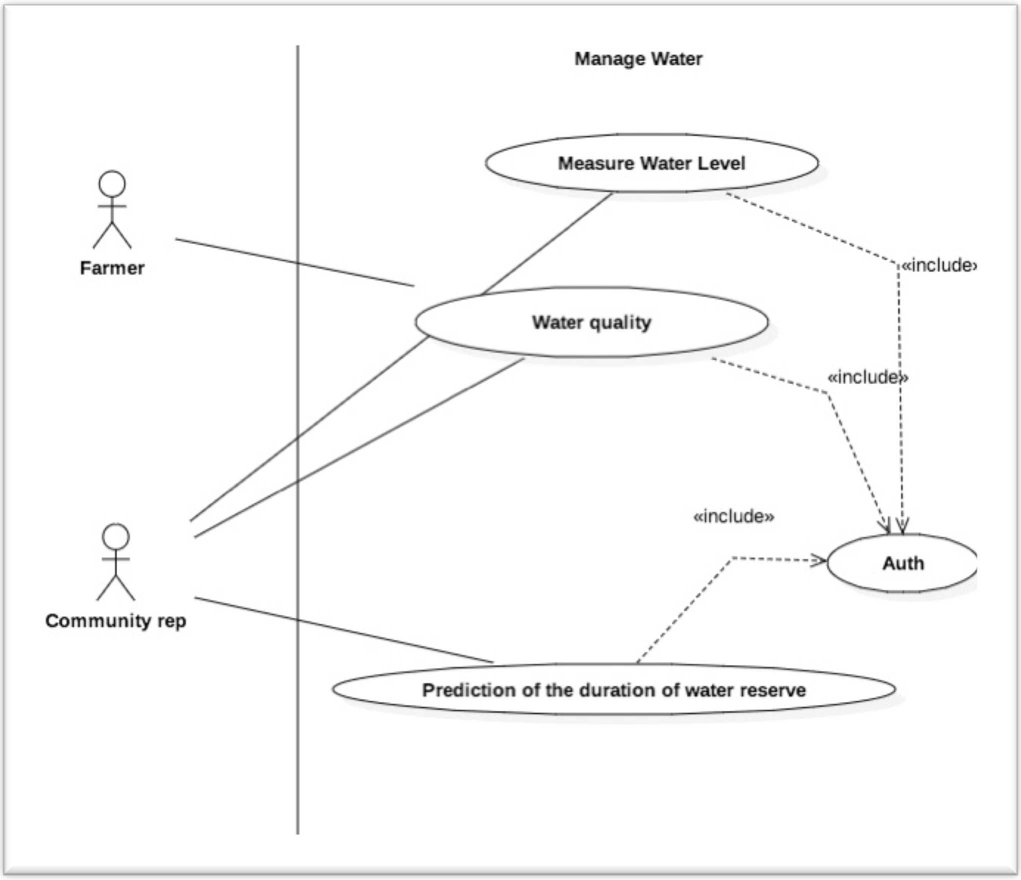


Figure 4: Water use case actor diagram

### 3.3.4 Use cases analysis

In the following each of the use cases is analysed with its potential for the business cases and also how the use case is mapped for WAZIUP's technical requirements.

**Table 18: Water use case analysis**

| Business Related Meta-Data | | | | | |
|---|---|---|---|---|---|
| **Uses Cases** | **B1 (end user impact)** | **B2 (service provider impact)** | **B3 (overall impact)** | **B4 (privacy)** | **B5 (actor)** |
| **UC1 Measure Water Level** | This use case will allow to select water quality as sensitivity is adjustable against various kinds of water | essential to know the evolution of water bodies and therefore that water reserves depending to weather conditions | Moderate | No privacy concern for the community representative | Community Representative |
| **UC2 Water quality** | This use will give valuable information to administrators, water managers and local people | Enable the provision of clean and safe water for human consumption, thereby protecting human health | Moderate | No privacy concern for famer/community representative | Farmer /Community Representative |
| **UC3 Prediction of the duration of water reserve** | This use case will help maintain the good quality water and almost all the indices satisfy the requirements | This use case will allow prediction to be more accurate and practical. It can provide better support of decision on management of water environment of the reservoir | Moderate | No privacy concern for community representative | Community Representative |

| Technical Meta-Data | | | | | | |
|---|---|---|---|---|---|---|
| **Uses Cases** | **T1 (Data Source)** | **T2 (User interaction)** | **T3 (IoT sensor)** | **T4 (Data analytic)** | **T5 (local and global)** | **T6 (infrastructure)** |
| **UC1 Measure Water Level** | This use case require sensors and gauges | SMS /USSD/web | Yes it relates to IoT | This use case can use Big data to allow prediction of availability | Local | Sensors, gateway + SMS based app / Web/Mobile App |
| **UC2 Water** | This use case | SMS | Yes it relates | This use case | Local | Sensors, |

| | | | | | | |
|---|---|---|---|---|---|---|
| **quality** | require sensors | /USSD/web | to IoT | can use Big data to allow prediction of quality | | gateway + SMS based app / Web/Mobile App |
| **UC3 Prediction of the duration of water reserve** | This use case require sensors and gauges | SMS /USSD/web | Yes it relates to IoT | This use case can use Big data to allow prediction and recommendation for water duration | Local | Sensors, gateway + SMS based app / Web/Mobile App |

## 3.4 Health

This section describes the use case and users stories, use case analysis of the health application domain.

### 3.4.1 Problem context

In a context of increased cost, complex medical demography and diversity of skills involved, eHealth vehicles the promise of better control of the health spending and more effective management of diseases. Of all the personal data we accumulate in our personal and digital lives, health data is one of the most sensitive categories. Inappropriate sharing of health information has the potential to damage careers, harm reputations and worse. To improve human health is the ultimate goal of any economic, technological and social development. The rapid rising and aging the population is one of the macro powers that will transform the world dramatically, it has caused great pressure to healthcare systems all over the world, and the emerging technology breakthrough of the Internet-of-Things (IoT) is expected to offer promising solutions.

### 3.4.2 User stories and use cases

In the following table, the user stories of the health domain are listed.

**Table 19: Health user stories**

| ID | User stories | Actors |
|---|---|---|
| **UC1** | As User I want to be able to know the available beds in a given Hospital in real time. | User |
| **UC2** | A doctor wants tools to perform a remote diagnose of a patient and assist to him. | Doctor |
| **UC3** | As customer I want to get emergency notifications from WAZIUP (epidemic risk, collective access point, etc.). | Patient |

| | | |
|---|---|---|
| **UC4** | As doctor I want to be able to track patient medications (current and past medications). | Doctor, patient |
| **UC5** | As customer I want to be able to get location and services of the nearest medical center. | Patient |
| **UC6** | As customer/patient I want to be able to get the nearest pharmacy and the drugs that I need. | Patient, doctor |
| **UC7** | As customer I want to be able to submit and track infant weight measure. | Doctor, customer |
| **UC8** | As consumer/Patent/Doctor I want to be able to access to the list of mandatory vaccines. | User, patient |
| **UC9** | As customer I want to track my vaccinations agenda. | Doctor |

#### *3.4.2.1 User case 1: Get number of beds available*

In the following table, the detailed description of the use case 1 is outlined.

**Table 20: Health use case 1**

| **Name** | **Get number of beds available** |
|---|---|
| **Descriptions** | User can know the number of available beds in a given hospital, Monitor health facility bed capacity and availability,<br>Provide real time occupancy ratio for emergency. |
| **Users** | User |
| **Preconditions** | Have sensors installed on hospital beds. |
| **Basic Course of Events** | 1. The User requests notification to know the available beds in the given hospital to WAZIUP through SMS/USSD<br>2. The system requests identity verification<br>3. Available bed data, possible actions and the process to follow are sent to the user |
| **Alternative Paths** | In step 2, if the authentication fails the system sends to the user the necessary information to register.<br>In step 3, If the bed data is not available, the system sends a notification asking user to try again later. |
| **Post conditions** | The update appears with most recent available bed data and the process to follow |

#### *3.4.2.2 User case 2: Remotely assist a patient living in an remote place*

In the following table, the detailed description of the use case 2 is outlined.

**Table 21: Health use case 2**

| **Name** | **Remotely assist a patient living in an remote place** |
|---|---|
| **Description** | A doctor can perform a distant diagnosis and give recommendations for medication and best practices to a patient living in a hard accessible place through the WAZIUP platform. The benefit is also to avoid moving a patient over a long distance.<br>Provides real time diagnosis tool in many formats (video, text, voice).<br>Free hand diagnosis tool for a doctor (no need to be at office to do his job).<br>Archive the stories as elements of medical data.<br>Reduce the occupancy of hospitals (hence the patient can stay in home care with his family). |
| **Users** | Doctor, patient |
| **Preconditions** | Doctor and patient both can be connected to WAZIUP (phone, PC, smartphone).<br>The doctor and the patient mutually agreed on the communication channel (video, text, voice) |
| **Basic Course of Events** | 1. The doctor and patient request a communication<br>2. WAZIUP request an identity checking<br>3. A communication is raised between the patient and his doctor |
| **Alternative Paths** | In step 2, if the authentication fails the system sends to the user the necessary information to register.<br>In step 3, If the communication is broken a notification is sent to the user asking him/her to try again. |
| **Post conditions** | The doctor and patient have access to data (video, text, voice)<br>The streaming data are stored on WAZIUP |

#### *3.4.2.3 Use case 3: Get emergency notifications.*

In the following table, the details description of the user story 3 is outlined.

**Table 22: Health use case 3**

| **Name** | **Get emergency notifications** |
|---|---|
| **Description** | Broadcast information on collective access point (ex. give the location of a place and the date of vaccinations, etc.), risk situation (epidemic, etc.).<br>Provide to population an emergency information-broadcasting tool |

| | |
|---|---|
| | during epidemic or other health risk situation. |
| **Users** | Customer |
| **Preconditions** | Customer has a required tool to receive WAZIUP alert.<br>Telecommunication services are available. |
| **Basic Course of Events** | 1. The WAZIUP platform sends alert to customer through SMS or mail format<br>2. The customer gets alert with SMS or in his/her mailbox<br>3. WAZIUP stops the broadcasting |
| **Alternative Paths** | 1. In step 1, if the platform failed sending alert a notification is sent to platform administrator. |
| **Post conditions** | The platform records information about alerts (sending time, quantity, etc.) |

#### *3.4.2.4 User case 4: Track a patient medication*

In the following table, the detailed description of the use case 4 is outlined.

**Table 23: Health use case 4**

| **Name** | **Track a patient medications** |
|---|---|
| **Description** | Track a patient's medications both in rural and urban area<br>Provide to a doctor the possibility to follow the medications of a patient (distant and near) and get the history of a patient<br>Provide a decisional tool to a doctor for medications prescription<br>Provide to a patient a history of his medications<br>Send to a patient an alert for taking medication |
| **Users** | Doctor |
| **Preconditions** | User has a required access to WAZIUP<br>The patient medications histories are registered on WAZIUP<br>WAZIUP platform is available |
| **Basic Course of Events** | 1. Doctor login on WAZIUP platform<br>2. Doctor selects a patient<br>3. Doctor submits or updates patient medications information (drugs, drugs allergic reactions, posology) |
| **Alternative Paths** | 1. In step 1 the doctor authentication failed or the attempt number is over; the system asks the doctor to renew his/her account or create a new one<br>2. In step 2 the doctor creates a new entry for patient |

| | |
|---|---|
| | 3. In step 3 the service is not available, the doctor logs out and registers the information in offline mode |
| **Post conditions** | The patient medications information is registered on WAZIUP platform |

#### *3.4.2.5 Use case 5: Get location and services in the nearest medical center*

In the following table, the details description of the use case 5 is outlined.

**Table 24: Health use case 5**

| **Name** | **Get location and services in the nearest medical center** |
|---|---|
| **Summary** | Get information on locations, service of the nearest medical center Provide to patients the possibility to locate in real-time the nearest medical center from his position and also give him a list of the different services available on the medical center. |
| **Users** | Patient |
| **Preconditions** | The system WAZIUP have through data entry all information about medical center and hospital (location, service) |
| **Basic Course of Events** | 1. Consumer login in to App or request info through his mobile and send his position<br>2. WAZIUP gets his position get the nearest medical center and other information.<br>3. Consumer receive the information on his app or mobile. |
| **Alternative Paths** | In step 1 the consumer authentication failed or the attempt number is over; the system asks the consumer to renew is account or create a new one |
| **Post conditions** | The patient receives the adequate information. |

#### *3.4.2.6 Use Case 6: Locate the nearest pharmacy selling the drugs a patient need*

In the following table, the details description of the use case 6 is outlined.

**Table 25: Health use case 6**

| **Name** | **Locate the nearest pharmacy selling the drugs a patient need** |
|---|---|
| **Description** | Send to patients the nearest pharmacy office selling specific drugs. Provide to patients the location of the nearest pharmacy selling the drug. |
| **Users** | Patient |

| | |
|---|---|
| **Preconditions** | Have a significant number of pharmacies mapped or with its coordinates available |
| **Basic Course of Events** | 1. User logs into WAZIUP (by web or USSD)<br>2. User sends a request for drug(s)<br>3. WAZIUP sends to user the nearest pharmacy or displays a mapped location of the nearest pharmacies |
| **Alternative Paths** | In step 1 the consumer authentication failed or the attempt number is over; the system asks the consumer to renew his/her account or create a new one |
| **Post conditions** | The user has the information based upon his request |

#### *3.4.2.7 Use case 7: Submit and track infant weight measure*

In the following table, the detailed description of the use case 7 is outlined.

**Table 26: Health use case 7**

| **Name** | **Submit and track infant weight measure** |
|---|---|
| **Description** | Help mothers follow their baby weight. It can be a good instrument for a doctor to provide advice to mothers.<br>Provide a tool to submit the weight of infant.<br>Provide a tool for pragmatic decision making. |
| **Users** | Patient, Doctor |
| **Preconditions** | Have a valid account on WAZIUP platform<br>Have valid information about infant in WAZIUP |
| **Basic Course of Events** | 1. User signs in to WAZIUP platform<br>2. User updates the infant weight measures<br>3. User requests information about an infant<br>4. WAZIUP provides weight measure |
| **Alternative Paths** | In step 1 the consumer authentication failed or the attempt number is over; the system asks the consumer to renew his/her account or create a new one<br>The user provides wrong information request; the system will send an empty result. |
| **Post conditions** | Infant weight measures are updated<br>WAZIUP provide valid weight measures to users |

#### *3.4.2.8 Use case 8: Access to the list of mandatory vaccines*

In the following table, the details description of the use case 8 is outlined.

Table 27: Health use case 8

| | |
|---|---|
| **Name** | **Access to the list of mandatory vaccines** |
| **Description** | Provide to user the mandatory vaccines based upon the age. Get access to the list of the mandatory vaccines.<br>Submit a mandatory vaccine to WAZIUP. |
| **Users** | Patient, Doctor |
| **Preconditions** | Mandatory vaccine lists have been saved to WAZIUP platform |
| **Basic Course of Events** | 1. User signs in to WAZIUP platform<br>2. User requests mandatory vaccine list.<br>3. Mandatory vaccines list is sent to user. |
| **Alternative Paths** | In step 1 the consumer authentication failed or the attempt number is over; the system asks the consumer to renew his account or create a new one.<br>At step 3 the system failed to get the vaccine list, will prompt user to try again later. |
| **Post conditions** | The up to date vaccine list is sent to the user. |

#### *3.4.2.9 Use Case 9: Track my vaccinations agenda*

In the following table, the detailed description of the use case 9 is outlined.

Table 28: Health use case 9

| | |
|---|---|
| **Name** | **Track my vaccinations agenda** |
| **Description** | Provide to customer/doctor an application to track vaccinations agenda. |
| **Users** | Patient, Doctor |
| **Preconditions** | User is registered to the vaccination agenda tracking system. |
| **Basic Course of Events** | 1. User is registered to the system<br>2. The system sends notification to user every time the vaccination schedule day and time approach. |
| **Alternative Paths** | In 2, if the system fails to send the notification, the manager will receive notification<br>Manager can manually send the notification to the user. |
| **Post conditions** | The user receives the day and time he needs to get his vaccines. |

### 3.4.3 Use case diagram and actor description

For health, nine use cases have been identified, with for main actors: Doctor, Patient and User, here User is the next of kin, who is accompanying the patient.

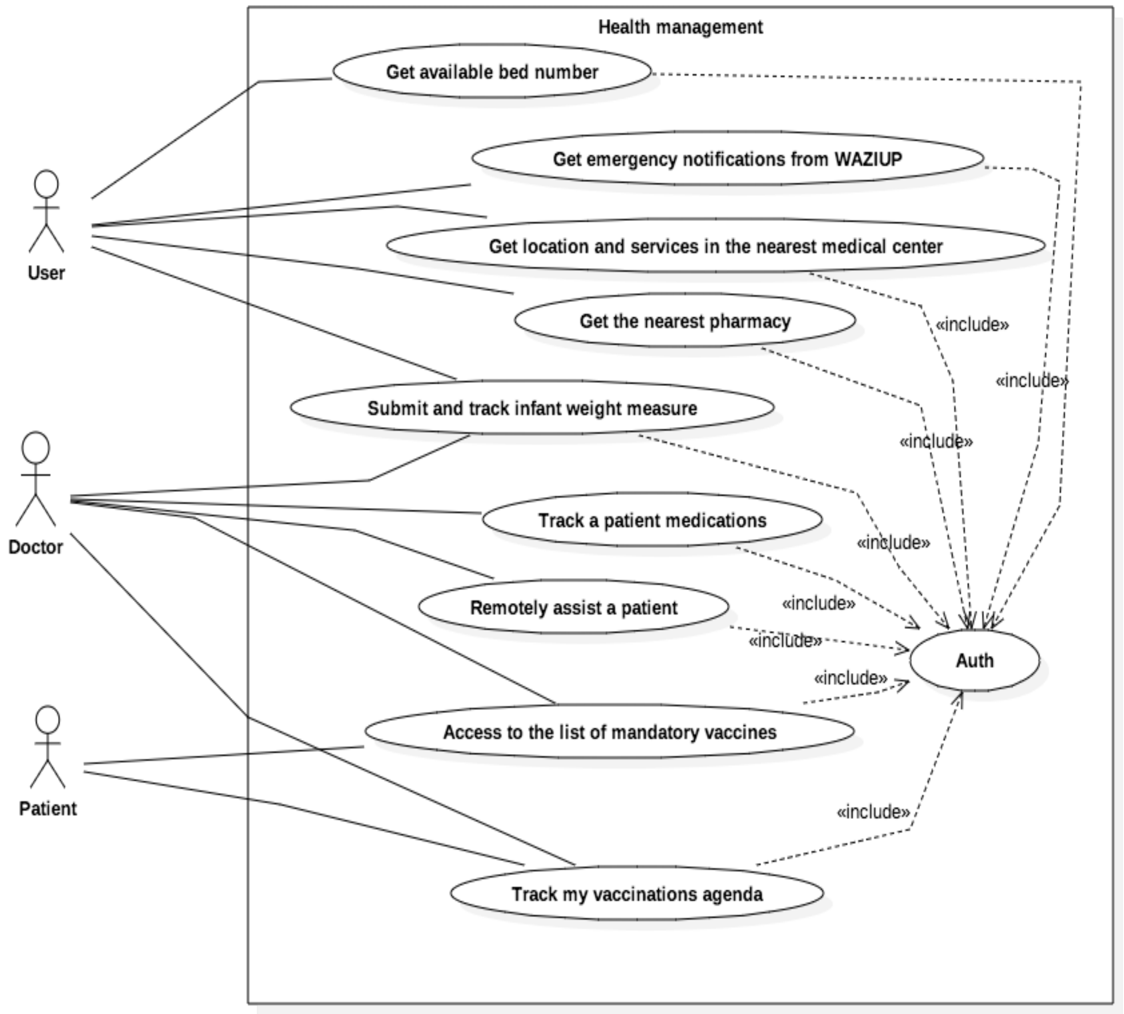


**Figure 5: Health use case actor diagram**

### 3.4.4 Use case analysis

In the following each of the use cases is analysed with its potential for the business cases and also how the use case map to WAZIUP technical requirements.

**Table 29: Health use case analysis**

| **Business Related Meta-Data** | | | | | |
|---|---|---|---|---|---|
| **Uses Cases** | **B1 (end user impact)** | **B2 (service provider impact)** | **B3 (overall impact)** | **B4 (privacy)** | **B5 (actor)** |
| **UC1 Get number of beds available** | Having smart bed that can detect when they are occupied, or when patient are trying to get up can be a real asset to healthcare improvement. | This use case will allow real time knowledge of the health facility bed-number. Most of African country counts bed availability through paper survey. | HIGH | This use case has some privacy concern, because of the data sensibility. | Patient, users, doctor |
| **UC2 Remotely assist a patient living in an remote place** | Reduce the distance from patient home to hospital. Reduce the risk of patient transportation. Gain of time and suppress transportation cost for patient and his Family. Helpful for the patient living in inaccessible or hard accessible location Share the expertise of a doctor to rural population. | Provide a medical platform tool to assist a patient anywhere Store the health information's of millions persons. | HIGH | Need for privacy concern since the platform will manage personal data from patient. Health data are very sensible an d critical. | Doctor, patient, researcher |
| **UC3 Get emergency notifications** | Reduce the cost of communication for epidemic alert. The population is informed in time about the risks of epidemic | Implement an alert system to broadcast the information. | HIGH | No privacy issue | Doctor, researcher, population, government |
| **UC4 Track a patient medications** | Reduce the risk of bad medication. Best decision for medical prescription. | Store the medications of millions of people. | MEDIUM | Privacy concern here since the medication history can reveal the health situation of a person | Government, Researcher, Patient Doctor |

| | | | | | |
|---|---|---|---|---|---|
| **UC5 Get location and services in the nearest medical center** | Reduce the cost and time for medical center search. Reduce the risk of mortality due to the lack of information on the location of medical center. | Store the location of medical centers. Map the max of the medical center. Implement a path short system | HIGH | No privacy | Government, Researcher, Patient Doctor |
| **UC6 Get the nearest pharmacy and the drugs a patient need** | Reduce the time and cost for the pharmacy and drug search. Reduce the mortality rate due to the lack of precision on pharmacy and drugs the patient needs | Store the location of pharmacy and the information on the drugs they sell. Map the pharmacies Implement the short path search system | HIGH | No privacy issue | Government, Researcher, Patient Doctor |
| **UC7 Submit and track infant weight measure** | Provide to mothers a tool to follow carefully the baby's weight. Help to take best decision to correct the health problem due to the weight issues. | Store the weight measures of millions of infants. Implement alert system | HIGH | Privacy issue | Government, Researcher, Patient Doctor |
| **UC8 Access to the list of mandatory vaccines** | Reduce the mortality rate due to the lack of information on vaccines. Excellent tool to sensitize the population to the vaccines | Store the list of mandatory vaccines. | HIGH | No privacy issue | Government, Researcher, Patient Doctor |
| **UC9 Track my vaccinations agenda** | Each citizen knows when he has to go to hospital for vaccine based upon the previous vaccinations and his age. Increase the number of vaccinated persons. | Store the list of mandatory vaccines. Store the vaccinations history of millions of persons. Implement agenda and predictive system | HIGH | Privacy need | Government, Researcher, Patient Doctor |

| **Technical Meta-Data** | | | | | | |
|---|---|---|---|---|---|---|
| **Uses Cases** | **T1 (Data Source)** | **T2 (User interaction )** | **T3 (IoT sensor )** | **T4 (Data analytic)** | **T5 (local and global)** | **T6 (infrastructur e)** |
| **UC1 Get number of beds available** | This use case need sensor. | SMS/Web/ Mobile web | Yes, it relates to IoT | Store the bed information in big data. | Local | Sensors, SMS based app/web/mobil e app |
| **UC2 Remotely assist a patient living in an remote place** | This use case need sensor. | SMS/Web/ Mobile web | Yes, it relates to IoT | Store the health information in big data. | Local | Sensors, SMS based app/web/mobil e app |

| | | | | | | |
|---|---|---|---|---|---|---|
| **UC3 Get emergency notifications** | Web, email or file from health institutions. | SMS/Web/ Mobile web | No | Store the information extracted from web, Implement Data mining. | Global | SMS based app/web/mobile app |
| **UC4 Track a patient medications** | File or user input. | SMS/Web/ Mobile web | No | Store the patient information, Implement machine learning for specific data extraction. | Local | SMS based app/web/mobile app |
| **UC5 Get location and services in the nearest medical center** | User input, Open source map platform. | SMS/Web/ Mobile web | No | Store the information extracted from web, Implement machine learning. | Local | SMS/Web/Mobile web |
| **UC6 Get the nearest pharmacy and the drugs a patient need** | User input, Open source map platform. | SMS/Web/ Mobile web | No | Store the information extracted from web, Implement machine learning. | Local | SMS/Web/Mobile web |
| **UC7 Submit and track infant weight measure** | User input, files. | SMS/Web/ Mobile web | No | Store the information extracted from web and from user input. | Local | SMS/Web/Mobile web |
| **UC8 Access to the list of mandatory vaccines** | User input, files. | SMS/Web/ Mobile web | No | Store the information. | Local | SMS/Web/Mobile web |
| **UC9 Track my vaccinations agenda** | User input, files. | SMS/Web/ Mobile web | No | Store the information of vaccinations. Implement predictive system. | Local | SMS/Web/Mobile web |

## 3.5 Environment

This section describes the use case and user's stories, use case analysis of the environment application domain.

### 3.5.1 Problem context

Pollution is one of the main environmental challenges that the world is facing today. The impact of pollution is more severe in developing countries, leading to bad health, death and disabilities of millions of people. However, while this may be easy for developed countries, halting environmental pollution may undermine economic growth and competitiveness of developing countries whose economies depends on natural resources.

Today, IoT-influenced technologies help from the foundation for environmental care. Industrial solutions let companies, as well as consumers, to play a significant role in controlling energy costs or monitoring local weather for example.

### 3.5.2 User stories and use cases

In the following table, the user stories of the environment domain are listed.

**Table 30: Environment user stories**

| ID | User stories | Actor |
|---|---|---|
| **UC1** | As security agency I want to be able to detect flood earlier as possible. | Security Agency |
| **UC2** | As Residential user I want to be able to monitor Air Quality | Residential user |
| **UC3** | As Waste manager I want to be able to confirm emptied waste bins | Waste Manager |

#### *3.5.2.1 Use case 1: Early Flood Detection.*

In the following table, the details description of the use case 2 is outlined.

**Table 31: Environment use case 1**

| **Name** | **Early Flood Detection** |
|---|---|
| **Description** | By implementing early flood detection, loss of lives can be reduced. Flood situations should be monitored at the earliest and send a notification in case of danger to WAZIUP platform. The notification sent to the platform should be accessible easily to end users who subscribe to the service. Using ultrasonic sensors connected to microcontrollers, water levels and value can be measured in dams and rivers that sends data/ information to the internet through WAZIUP IoT network.<br>Monitor water levels in (major) water bodies with relevant sensors (ultrasonic).<br>Send water level (raw) data / information to WAZIUP. |
| **Users** | Security Agency |
| **Preconditions** | Sensors must be placed and installed in the water bodies and connected to |

| | |
|---|---|
| | the IoT network. Application developers must subscribe to the information sent to WAZIUP. |
| **Basic Course of Events** | 1. Sensors must periodically send data / information to the IoT network<br>2. Data / information is added to the Big Data on WAZIUP<br>3. Application developers request and pull data from WAZIUP (per the maximum requests per seconds).<br>4. End user in this case: Waste Manager, Residential User, Security Agency, Disaster Prevention Agency requests for data via SMS / USSD / GPRS. |
| **Alternative Paths** | In step 3, Application developer cannot request for the data/information that he has not subscribed to or which have been made available by the sensor owner.<br>In step 4, Only available data/information can be made available on request |
| **Post conditions** | End users, will be made available of the water levels and recommendations |

#### *3.5.2.2 Use case 2: Monitor Air Quality*

In the following table, the details description of the use case 2 is outlined.

**Table 32: Environment use case 2**

| **Name** | **Monitor Air Quality** |
|---|---|
| **Description** | A residential user would want to know the air quality in and outside his house. They also want to receive notification if the air quality is below a critical level:<br>1. Monitor air quality (NO2 and CO) indoor and outdoor<br>2. Aggregate air quality data<br>3. Update user on air quality |
| **Users** | Residential user |
| **Preconditions** | Air quality sensor must be installed and connected to WAZIUP |
| **Basic Course of Events** | 1. Sensors are powered either indoor or outdoor to send data to WAZIUP<br>2. WAZIUP stores and makes the data available to subscribers<br>3. Subscribers and application developers make information available to users on request |
| **Alternative Paths** | In step 3, Application developer cannot request for data / information he has not subscribed to or which have been made available by the sensor owner. |
| **Post conditions** | Air quality information is made available and updated in real time and recommendations |

#### 3.5.2.3 Use case 3: Confirm emptied waste bins

In the following table, the details description of the use case 3 is outlined.

**Table 33: Environment use case 3**

| Name | Confirm emptied waste bins |
|---|---|
| Description | The waste manager wants to confirm if waste bins have been lifted and emptied by the compactor trucks:<br>1. Test for weight and level of waste bin<br>2. Upload data together with the time action took place |
| Users | Waste Manager |
| Preconditions | Sensors installed and connected to WAZIUP |
| Basic Course of Events | 1. Sensor measures level of bin during the time it lifts and empties bin<br>2. Sensor uploads data to WAZIUP<br>3. Data is added to Bid Data for use by sensor owner |
| Alternative Paths | If sensor owner does not allow for public subscription data is available privately |
| Post conditions | Later updates are sent to data subscribers |

### 3.5.3 Use case diagram and actor description

Five use cases have been identified for environment with four main actors: residential user, waste manager, security agency.

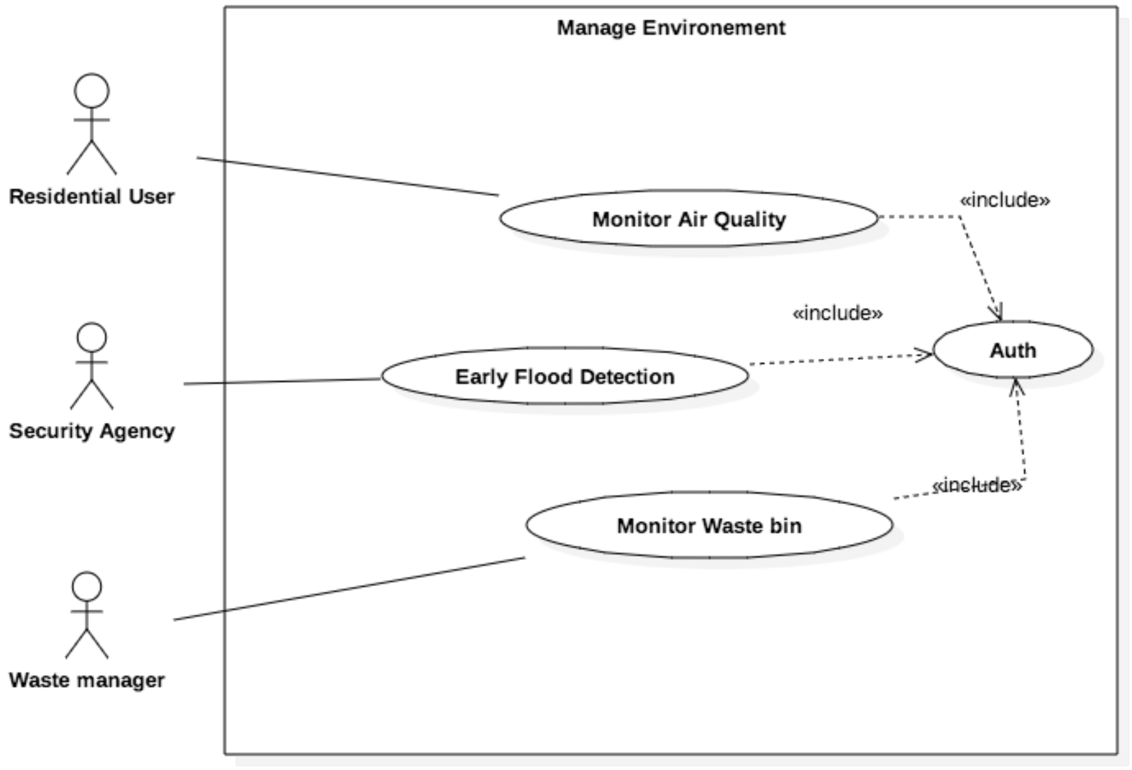


Figure 6: Environment use case actor diagram

### 3.5.4 Use case analysis

In the following each of the use case is analyzed with its potential for the business cases and also how the use case mapped to WAZIUP technical requirements.

Table 34: Environment use case analysis

| **Business Related Meta-Data** | | | | | |
|---|---|---|---|---|---|
| **Uses Cases** | **B1 (end user impact)** | **B2 (service provider impact)** | **B3 (overall impact)** | **B4 (privacy)** | **B5 (actor)** |
| **UC1 Early Flood Detection** | Receiving early warning notification about possible flood can help App user take certain action in preventing the water from affecting their houses. | By tracking data like rainfall, ground saturation, and upstream water levels, Service provider can get early warnings about flooding and take action | Medium | This may have some legal and regulatory issue since water quality is directly related to public health | Service provider, Population |
| **UC2 Monitor Air Quality** | Citizens will directly benefit from Air | Monitor pollution levels in central cities | One of the Africa's facts the | This may have some legal and | Citizen, Service |

| | | | | | |
|---|---|---|---|---|---|
| | pollution monitoring by getting adequate information about indoor and outdoor information on the air quality. | are key to provide adequate information to citizens and take actions to reduce it. | lack of air measurements and the lack of medical studies linking pollution to deaths in Africa. So Air pollution can have HIGH impact | regulatory issue since water quality is directly related to public health | provider. |
| **UC3 Confirm emptied waste bins** | Citizen wants to know when compactor truck empty community waste bins, this will help people change their behaviour, by so avoiding overloaded bins. | Service provider can benefit from form this use case to know which bins are full and need to be emptied. | Medium | | Home owner, service provider |

| **Technical Meta-Data** | | | | | |
|---|---|---|---|---|---|
| **Uses Cases** | **T1 (Data Source)** | **T2 (User interaction)** | **T3 (IoT sensor)** | **T4 (Data analytic)** | **T5 (local and global)** |
| **UC1 Early Flood Detection** | This UC requires an water detection sensor. | Through a mobile app. | Yes it relates to IOT. | Big data allows the security agent to know in advance if a flood is coming. | Global |
| **UC2 Monitor Air Quality** | Data collected through a sensor. | Through a mobile app. | Yes, relates to IOT. | Big data allows everyone to be aware of the air quality and be alerted if it goes below a critical point. | Global |
| **UC3 Confirm emptied waste bins** | Data collected through sensor. | Through a mobile app. | Yes, relates to IOT. | | |

## 3.6 Logistics and transport

This section describes the use case and user's stories, use case analysis of the logistic and transports application domain.

### 3.6.1 Problem context

Logistic and Transport are fundamentally about moving things from one place to another. Therefore, the main service components of Logistic and Transport can be categorized into the things that move and the things that do the moving—the "demand" and "supply" sides of logistics. Whether by air, ground or sea, transportation and logistics are essential components to many enterprises' productivity, and access to real-time data is critical. Many industries and business sectors are struggling to grasp the possibilities of data-driven technology, but companies in transport and logistics are way ahead. By their very nature, the logistics providers that move objects by air, sea, rail, and ground have widely distributed networks and rely on rapid information about those networks to make decisions. As a result, they were quick to see the benefits of new sensor and connection technology, placing them at the forefront of the transition to a connected world.

### 3.6.2 User stories and use cases

In the following table, the user stories of the logistics and transport domain are listed.

**Table 35: Logistic and transport user stories**

| ID | User stories | Actors |
|---|---|---|
| **UC1** | As operator I want to be able to track, operations, and remote monitoring | Operator |
| **UC2** | As operator I want to be able to get real-time visibility across the supply chain | Operator |
| **UC3** | As manager I want to be able to check the integrity, identification, authentication and traceability | Manager |

#### *3.6.2.1 Use case 1: Track operations and remote monitoring*

In the following table, the details description of the user story 1 is outlined.

**Table 36: Logistic and transport use case 1**

| **Name** | **Track operations and remote monitoring** |
|---|---|
| **Description** | The connected object may contain information on its own history: where do the raw materials, which were there assembled, how and under what conditions he has been transported |
| **Users** | Operator |
| **Preconditions** | Customer has devices able to connect to WAZIUP. |
| **Basic Course of Events** | 1. Allow workers to use SMS to report slow-moving or low inventory<br>2. Use SMS to enable workers to log hours (for example, the beginning and end of shifts) |

| | |
|---|---|
| | 3. Use “wake-up” SMS to enable data connections for remote M2M devices |
| **Alternative Paths** | Machine-to-machine (M2M) remote monitoring applications can trigger SMS alerts automatically |
| **Post conditions** | Notification about tracking |

#### *3.6.2.2 Use case 2: Real-time visibility across the supply chain*

In the following table, the details description of the use case 2 is outlined.

**Table 37: Logistic and transport use case 2**

| **Name** | **Real-time visibility across the supply chain** |
|---|---|
| **Description** | Fulfil services providers with real time location and messaging capabilities to optimize and streamline operations. |
| **Users** | Operator |
| **Preconditions** | Worker have device connect on WAZIUP application |
| **Basic Course of Events** | 1. Reduce failed delivery attempts by using automated SMS delivery updates<br>2. Reduce calls to care centers using SMS proactively to manage changes<br>3. Use SMS to delivery by using keywords, such as “DELIVERED”<br>4. Send SMS alert such as “Your delivery is scheduled for [date/time]<br>5. Use SMS with a URL to invite customers to confirm change location, etc. |
| **Alternative Paths** | If the authentication fails, the system send to the user necessary information to register |
| **Post conditions** | Worker have all information for the visibility across the supply chain |

#### *3.6.2.3 Use case 3: Check the integrity, identification, authentication and traceability of goods*

In the following table, the details description of the use case 3 is outlined.

**Table 38: Logistic and transport use case 3**

| **Name** | **Check the integrity, identification, authentication and traceability of goods** |
|---|---|
| **Summary** | Process of recognizing the attributes that identify the object. Have a process of corroborating an entity or attributes with a specified or understood level of assurance. |
| **Users** | Manager |

| | |
|---|---|
| **Preconditions** | Manager/customer have access to WAZIUP application |
| **Basic Course of Events** | 1. Allow to recognize the identity and authenticity of the products<br>2. Authentication is correlated with the identification in order to initiate the ad 'hoc process |
| **Alternative Paths** | If step 1 fails, the system send to the user necessary information to register. |
| **Post conditions** | Manager receive all data about products for integrity, identification, authentication and traceability. |

### 3.6.3 Use case diagram and actor description

We have 3 uses case for logistic and transport. Two actors, worker and customer manager shared those use cases. Worker tracks operation and remote monitoring and has a real time visibility. Customer manager has to verify the identification and integrity.

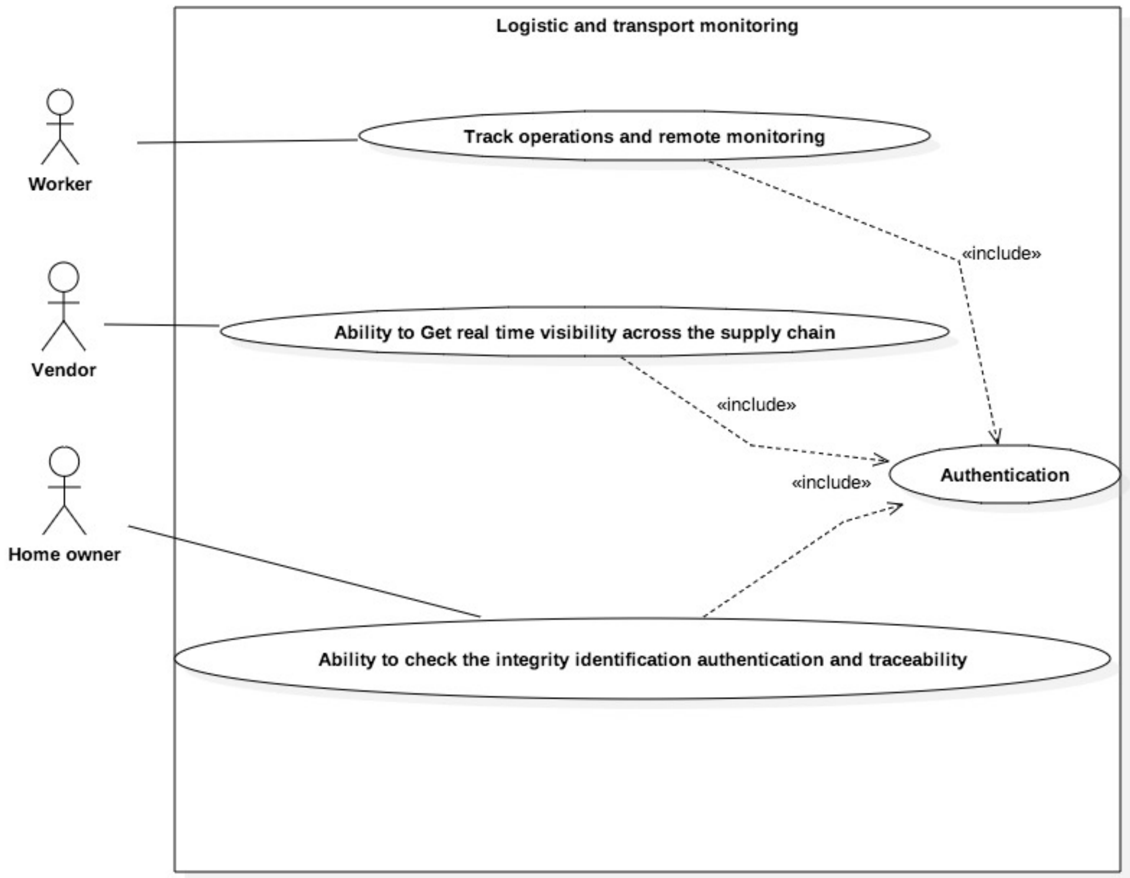


Figure 7 : Logistic and transport use case actor diagram

### 3.6.4 Use case analysis

In the following tables each of the use case is analyzed with its potential for the business cases and also how the use case mapped to WAZIUP technical requirements.

**Table 39: Logistics and transport use case analysis**

| **Business Related Meta-Data** | | | | | |
|---|---|---|---|---|---|
| **Uses Cases** | **B1(end user impact)** | **B2 (service provider impact)** | **B3 (overall impact)** | **B4 (privacy)** | **B5 (actor)** |
| **UC1 Track, operations, and remote monitoring** | This use case will allow customer to track and monitor from one end to the route of the goods. | This use case will allow workers to track and monitor from one end to the route of the goods. | High | Yes, it relates to IoT | Worker |
| **UC2 Real-time visibility across the supply chain** | This use case will allow to have real-time visibility across the supply chain. | The service provider will be more effective.<br>The service provider will get ready for real-time decision making with the ability to swiftly adapt your execution across the end-to-end supply chain | High | Yes it relates to IoT | Worker/manager |
| **UC3 Ability to check the integrity, identification, authentication and traceability** | This use case will allow the customer to be only authorized persons have access to resources and ensure that the data are those believed to be | This use case will allow the service provider to have a control of the products according to regulations. | High | Yes it relates to IoT | Manager/worker/customer |

| **Technical Meta-Data** | | | | | | |
|---|---|---|---|---|---|---|
| **Uses Cases** | **T1 (Data Source)** | **T2 (User interaction)** | **T3 (IoT sensor)** | **T4 (Data analytic)** | **T5 (local and global)** | **T6 (infrastructure)** |
| **UC1 Track, operations, and remote** | This use case require sensors to track and | SMS/Web/Mobile web | Yes, it relates to IoT | Prediction, forecasting | Local | sensors, SMS based app/web/mo |

| | | | | | | |
|---|---|---|---|---|---|---|
| **monitoring** | monitor. | | | | | bile app |
| **UC2 Real-time visibility across the supply chain** | This use case require sensors to have real-time location. | SMS/Web/Mobile web | Yes, it relates to IoT | Prediction of the situation, Forecasting | Local | sensors, SMS based app/web/mobile app |
| **UC3 Ability to check the integrity, identification, authentication and traceability** | This use case require sensor to verify integrity, identification, authentication and traceability. | SMS/Web/Mobile web | Yes, it relates to IoT | Recommendation, prediction of the situation | Local | sensors, SMS based app/web/mobile app |

## 3.7 Urban and Domestic

This section describes the use case and user's stories, use case analysis of the urban and domestic application domain.

### 3.7.1 Problem context

Today, African cities are the one urbanizing most quickly and where will arise in the very near horizon, the problem of the availability of resources. Food problem which is especially threatening the momentum of our cities is linked to an exodus draining continuously rural living force; the same one who in the old system, had only function of feeding the cities. This is already present with the use of junk food, industrial products from distant sources and the traceability that is problematic. There is however a potential in some cities like Lomé, less battered than Lagos where we still have a real park (wild dumps, not built sites, abandoned lands, floodplains) that could be used for solving this problem. As already many market gardeners in our cities and all around are getting better structured and try to rethink how to face the challenges.

The IoT can conveniently help to reinforce this proximity activity with the use of more smart approaches based on ICT. Indeed, Urban agriculture is the playground for the connected devices, not only because of the presence of infrastructure. Connected agriculture also supports a specific question to the urban context, such as pollution (air, soil heavy -metals), lack of privacy and face, the rush situations (shorter time in town) the lack of empirical and non-dedicated population.

### 3.7.2 User stories and use cases

In the following table, the user stories of the logistics and transport domain are listed.

**Table 40: Urban and domestic user stories**

| ID | User story | Actors |
|---|---|---|
| **UC1** | As homeowner I want to have automation for my indoor/small scale farming. | Homeowner |
| **UC2** | As home owner I want to be able to save Energy | Homeowner |

| | | |
|---|---|---|
| **UC3** | As vendors/Clients I want to be able to locate street vendors | Vendors - Clients |
| **UC4** | As Residential user I want to be able to monitor Septic Tank Level | Homeowner |

#### *3.7.2.1 Use case 1: Indoor/small scale farming automation.*

In the following table, the details description of the use case 1 is outlined.

**Table 41: Urban and domestic use case 1**

| **Name** | **Indoor/small scale farming automation** |
|---|---|
| **Description** | Automating the daily actions for indoor farming. (E.g., it has rained; automatic watering therefore not be triggered). Dynamic Regulations of sunshine, light, temperature, humidity, gasses, watering automation by reacting to changes in data delivered by a platform. |
| **Users** | Individuals, producers |
| **Preconditions** | WAZIUP capacity to compile and organize data. Connect all intelligent devices to WAZIUP |
| **Basic Course of Events** | 1. The device is permanently connected to WAZIUP<br>2. It reacts to fluctuation thresholds on WAZIUP data (e.g. temperature) |
| **Alternative Paths** | In step 2, Device may be offline or not receive data. It should idle until it get back online. |
| **Post conditions** | Device reacts accordingly to data from sensors. |

#### *3.7.2.2 Use case 2: House energy saving*

In the following table, the details description of the use case 2 is outlined.

**Table 42: Urban and domestic use case 2**

| **Name** | **House energy saving** |
|---|---|
| **Description** | I would like the lights switch off in a room as soon as it is empty, under a lamppost as soon as there is no one passing in the street. Ability to detect the absence of movement for a period of time so as to extinguish any object using electricity. I get an alert as soon as a space is empty and on. |
| **Users** | Administrations, facilities, public places in which no one feels really concerned with energy conservation. |
| **Preconditions** | The sensors installed in the affected parts and connected lighting to my cell phone so I can turn off the remote parts. |

| | |
|---|---|
| **Basic Course of Events** | 1. The user send notification request WAZIUP through SMS / USSD / Web<br>2. The request identity verification system<br>3. The user sends credential.<br>4. The presence sensors signal the presence or absence of people in the room.<br>5. The system sends directly to the connected lamps the indications so that they go off. |
| **Alternative Paths** | In step 2, if the authentication fails the system send to the user necessary information to register.<br>In step 5, if the connection between the connected object and the oven does not work, the system sends to the user information about the presence or absence of people in the room so he can set the lights off. |
| **Post conditions** | User has direct control of electronic device energy consumption. |

#### *3.7.2.3 Use case 3: Geolocation of street vendors*

In the following table, the details description of the use case 3 is outlined.

**Table 43: Urban and domestic use case 3**

| **Name** | **Geolocation of street vendors** |
|---|---|
| **Description** | I want to find a hairdresser, shoemaker, etc.<br>A street vendor wants to be easily found. |
| **Users** | Sellers and buyers |
| **Preconditions** | Sellers have a connected device that geotagging them and send their course and location to the platform. They can also inform their location on a map where geolocation would not work. |
| **Basic Course of Events** | 1. The user sends notification request WAZIUP through SMS / USSD / Web<br>2. The system requests identity verification<br>3. The user sends credential.<br>4. The user can view on a map the location of the seller or seller type he seeks. |
| **Alternative Paths** | In step 2, if the authentication fails the system sends to the user Necessary information to register.<br>In step 4, if the system finds no craftsman sought by the client, it advises him/her to try again later or offers to receive notification when the person sought is found. |
| **Post Conditions** | The user no longer needs to wait for the chance meeting to satisfy a need.<br>The seller knows how to find. |

#### 3.7.2.4 Use case 4: Septic Tank Monitor

In the following table, the details description of the use case 4 is outlined.

Table 44: Urban and domestic use case 4

| Name | Septic Tank Monitor |
|---|---|
| Descriptions | Homeowner wants to understand how their septic tanks work and their capacity limitation. The property owner or occupants should be alerted when their septic tanks are full. |
| Features | 1. Monitor septic tank capacity level<br>2. Send data on capacity level to WAZIUP<br>3. Alert homeowner when septic waste exceeds maximum level |
| Users | Homeowner |
| Preconditions | Sensors must be installed and connected to WAZIUP |
| Basic Course of Events | 1. Sensor periodically monitors water levels<br>2. Sensor uploads data to WAZIUP<br>3. Data is added to Big Data for use by sensor owner |
| Alternative Paths | If sensor owner does not allow for public subscription data is available privately |
| Post conditions | Homeowner continues to monitor sewage levels and is notified before it reaches an alarming level. |

### 3.7.3 Use case diagram and actor description

Urban and domestic contains 4 use cases with 3 actors. Figure 7 depicts the use case diagram, representing the different actors and specific use cases.

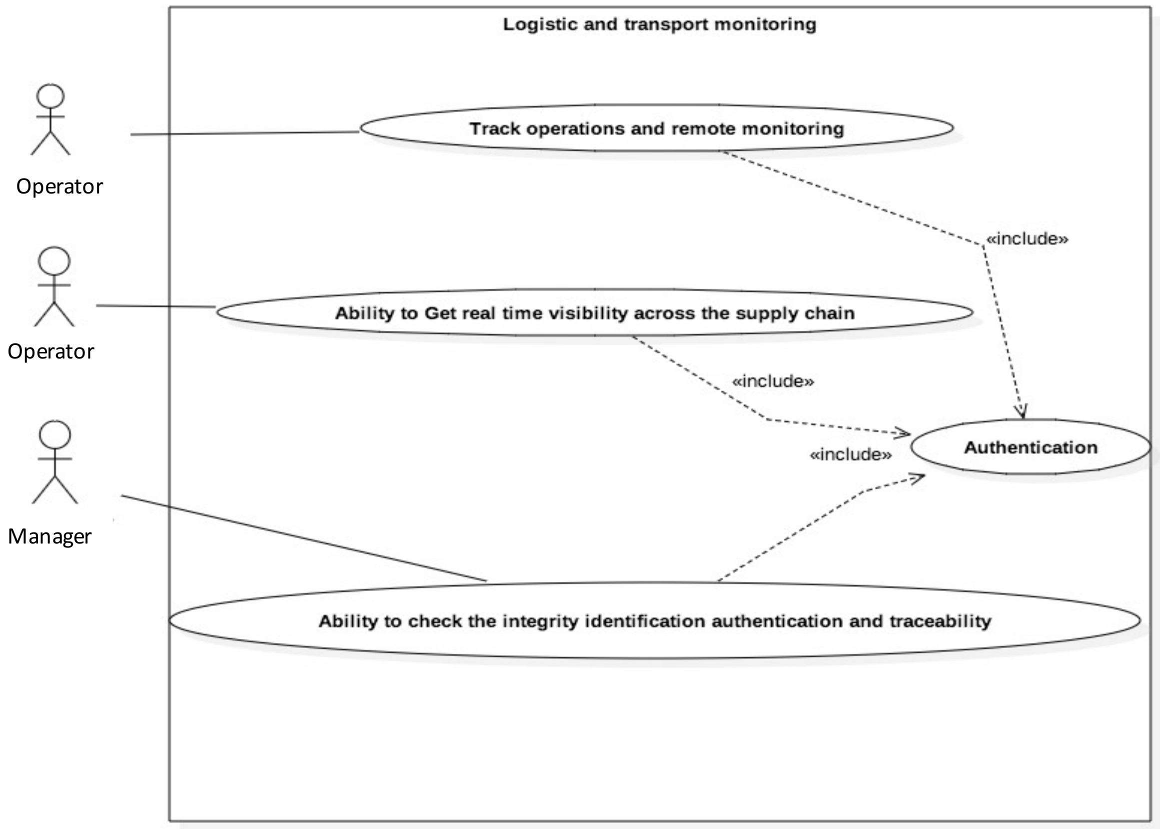


Figure 8: Urban and domestic use case actor diagram

### 3.7.4 Use case analysis

In the following table each of the use case is analyzed with its potential for the business cases and also how the use case mapped to WAZIUP technical requirements.

Table 45: Urban and domestic use case analysis

| Business Related Meta-Data | | | | | |
|---|---|---|---|---|---|
| **Uses Cases** | **B1 (end user impact)** | **B2 (service provider impact)** | **B3 (overall impact)** | **B4 (privacy)** | **B5 (actor)** |
| **UC1 Indoor/small scale farming automation** | This use case will allow to develop agriculture in the frame of the interface and interior culture | Give intention to automate processes, like the illumination and remote control of the home environment equipment. | Moderate | Privacy concern | Individuals/producers |

| | | | | | |
|---|---|---|---|---|---|
| **UC2 House energy saving** | This use case allows to reduce the amount of energy required to provide products and services. | Give a way of managing and restraining the growth in energy consumption. | Moderate | Privacy concern | Individuals, producers |
| **UC3 Geolocation of street vendors** | This use case gives information associated with an electronic device that can be used to identify its physical location. | Give providers methodologies for determining users locations | Moderate | Privacy concern | Sellers/buyers |
| **UC4 Septic Tank Monitor** | Homeowner wants to understand how their septic tanks work and their capacity limitation. The property owner or occupants should be alerted when their septic tanks are full. | Service provider can benefit from Septic tank monitoring data to provide service to homeowner. | Moderate | This may have some legal and regulatory issue since water quality is directly related to public health | Homeowner Service provider |

| **Technical Meta-Data** | | | | | | |
|---|---|---|---|---|---|---|
| **Uses Cases** | **T1 (Data Source)** | **T2 (User interaction)** | **T3 (IoT sensor)** | **T4 (Data analytic)** | **T5 (local and global)** | **T6 (infrastructure)** |
| **UC1 Indoor/small scale farming automation** | This use case requires sensors. | Web | Yes it relates to IoT. | Prediction of status | Local | Sensors, Web |
| **UC2 House energy saving** | This use case requires sensors. | SMS based push service. | Yes it relates to IoT. | Prediction of the situation | local | Sensors, Web |
| **UC3 Geolocation of street vendors** | This use case requires sensors. | SMS/USSD/web. | Yes it relates to IoT. | Data analytic | local | Sensors/web |
| **UC4 Septic Tank Monitor** | Data collected through sensor. | Through a mobile app. | Yes it relates to IoT. | Data analytic | local | Sensors |

## 3.8 Cattle rustling

This section describes the use case and user's stories, use case analysis of the cattle rustling application domain.

### 3.8.1 Problem context

Cattle rustling, a catch-all term which includes the pillaging and pilfering of cattle, sheep, goats and camel is pervasive throughout Africa. Nomadic movements of people characterize the region with their livestock on vast and hostile terrains with inadequate physical and communication infrastructure. The vice has caused havoc to many inhabitants of the region and has been the main source of conflict and instability. No adequate mechanism is in place to aid in identification and recovery of stolen animals, which is necessary in addressing these challenges.

### 3.8.2 User stories and use cases

In the following table, the user stories of the logistics and transport domain are listed.

**Table 46: Cattle rustling user stories**

| Use case ID | Description | Actors |
|---|---|---|
| **UC1** | As farmer I would like to have the ability to get real-time position and itinerary of the cattle's herd. | Farmer |
| **UC2** | As farmer I would like to have the ability to receive notification in critical situations. | Farmer |

#### 3.8.2.1 Use case 1: Real-time position and itinerary of the cattle's herd

In the following table, the details description of the use case 1 is outlined.

**Table 47: Cattle rustling use case 1**

| **Name** | **Real-time position and itinerary of the cattle's herd** |
|---|---|
| **Description** | As farmer I would like to have the ability to get real-time speed, position and itinerary of the cattle's herd |
| **Users** | Farmer |
| **Preconditions** | Sensors have been installed on animal |
| **Basic Course of Events** | 1. The farmer request notification to WAZIUP through SMS /USSD<br>2. The system request identity verification<br>3. The user sends credential.<br>4. Cattle speed, position and itinerary data and possible actions against it are sent to the user |

| | |
|---|---|
| **Alternative Paths** | In step 2, if the authentication fails the system send to the user necessary information to register.<br>In step 4, If the data, the system send notification asking user to try again later. |
| **Post conditions** | User is updated with most recent data and can take action. |

#### *3.8.2.2 Use case 2: Get notification in critical situations.*

In the following table, the details description of the use case 2 is outlined.

**Table 48: Cattle rustling use case 2**

| **Name** | **Get notification in critical situations** |
|---|---|
| **Description** | As farmer I would like to have the ability to receive notification in critical situations |
| **Users** | Farmer |
| **Preconditions** | Sensors have been installed on animal |
| **Basic Course of Events** | 1. The farmer is registered to WAZIUP platform<br>2. Whenever a critical situation is observed (climate, disease, …) notification and possible actions against it are sent to the user. |
| **Alternative Paths** | No |
| **Post conditions** | User is updated with most recent information and can take action. |

### 3.8.3 Use case diagram and actor description

Two use cases have been identified for cattle rustling. Figure 8 depicts the interaction between the actor and the two use cases above.

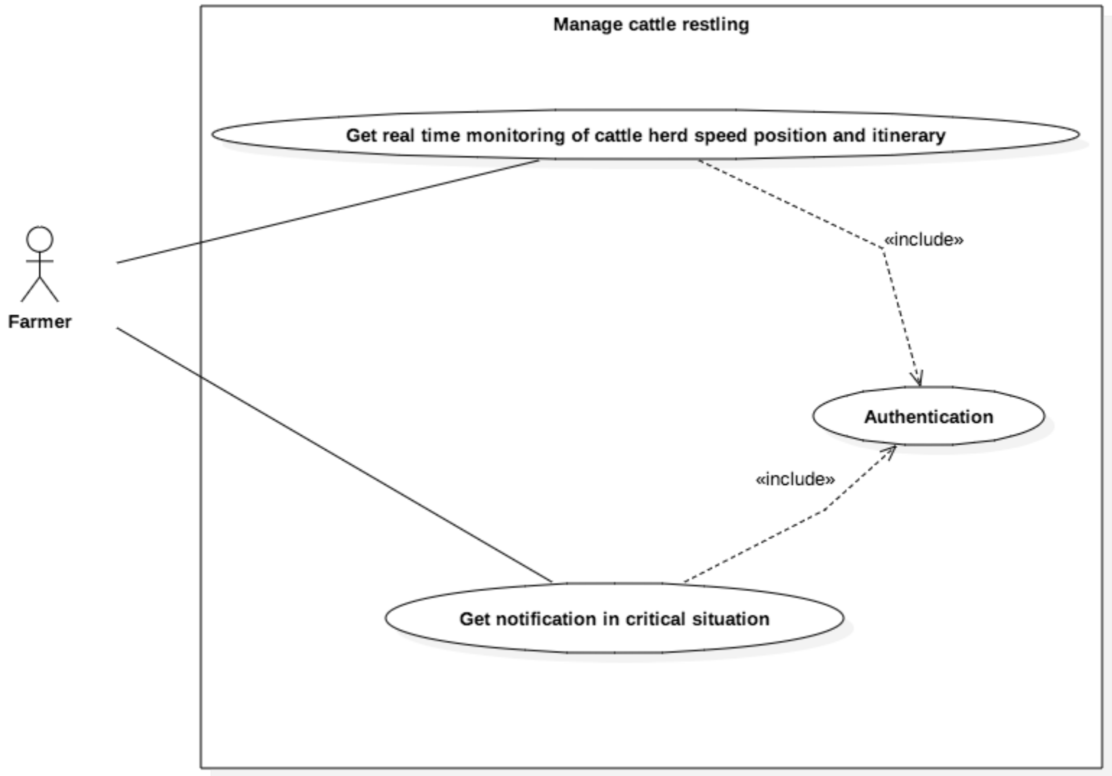


Figure 9: Cattle rustling use case actor diagram

### 3.8.4 Use case analysis

In the following each of the use case is analysed with its potential for the business cases and also how the use case map to WAZIUP technical requirements.

Table 49: Urban and domestic use case analysis

| **Business Related Meta-Data** | | | | | |
|---|---|---|---|---|---|
| **Uses Cases** | **B1 (end user impact)** | **B2 (service provider impact)** | **B3 (overall impact)** | **B4 (privacy)** | **B5 (actor)** |
| **UC1 Real-time position and itinerary of the cattle's herd** | This use case will allow farmers to have real time monitoring and have position and itinerary of their beasts. | Give a process through which farmers can evaluate use of data on software. It enables farmers to review the overall processes on the data in real time, or as it happens. | High | No privacy concern | Farmer |
| **UC2 Get notification in critical situations** | This use case will allow farmers to get | Automatically alert farmers of any potential | High | No privacy concern | Farmer |

| | | | | | |
|---|---|---|---|---|---|
| | notification in critical situation like theft. | threats. Gives farmers the ability to act quickly and smartly before an emergency situation even happens. | | | |

| Technical Meta-Data | | | | | | |
|---|---|---|---|---|---|---|
| **Uses Cases** | **T1 (Data Source)** | **T2 (User interaction )** | **T3 (IoT sensor)** | **T4 (Data analytic)** | **T5 (local and global)** | **T6 (infrastructure)** |
| **UC1 Real-time position and itinerary of the cattle's herd** | This use case require sensor to get localization data, speed traceability. | SMS/Web/ Mobile web | Yes, it relates to IoT | Recommendation, prediction of the situation | Local | sensors, SMS based app/web/mobile app |
| **UC2 Get notification in critical situations** | This does not directly relate to sensors, but infrastructure is needed to send SMS to farmer. | SMS/Web/ Mobile web | No | Recommendation, prediction of the situation | Local | SMS based app/web/mobile app |

## 3.9 Summary of the use cases

In this section we presented 31 use cases. They are summed up in the Table 50.

**Table 50: Domains and use cases**

| Domains | Use cases |
|---|---|
| Agriculture | UC1 Monitor soil |
| | UC2 Monitor Field Weather Data |
| | UC3 Storage Moisture and Temperature |
| Fish farming | UC1 Fish Pond Water Quality |
| | UC2 Estimate Fish Quantity |
| | UC3 Cost-Efficient Feeding System |
| | UC4 Plankton measurement for fish feeding supply accuracy |
| Water | UC1 Measure Water Level |
| | UC2 Water quality |
| | UC3 Prediction of the duration of water reserve |
| Health | UC1 Get number of beds available |

| | |
|---|---|
| | UC2 Remotely assist a patient living in an remote place |
| | UC3 Get emergency notifications |
| | UC4 Track a patient medications |
| | UC5 Get location and services in the nearest medical center |
| | UC6 Get the nearest pharmacy and the drugs a patient need |
| | UC7 Submit and track infant weight measure |
| | UC8 Access to the list of mandatory vaccines |
| | UC9 Track my vaccinations agenda |
| **Environment** | UC1 Early Flood Detection |
| | UC2 Monitor Air Quality |
| | UC3 Confirm emptied waste bins |
| **Logistics and transport** | UC1 Track operations and remote monitoring |
| | UC2 Real-time visibility across the supply chain |
| | UC3 Check the integrity, identification, authentication and traceability of goods |
| **Urban and Domestic** | UC1 Indoor/small scale farming automation |
| | UC2 House energy saving |
| | UC3 Geolocation of street vendors |
| | UC4 Septic Tank Monitor |
| **Cattle rustling** | UC1 Real-time position and itinerary of the cattle's herd |
| | UC2 Get notification in critical situations |

# 4 USERS FEEDBACK ANALYSIS

Using the survey results and partner's choice about the use cases, in addition to the use cases analysis matrix, we did a table ranking of the use cases. This allows us to have a better view on the use cases that are more likely to have a great impact from an end user point of view as well as the Service provider point of view.

## 4.1 Agriculture

Agriculture uses cases Survey have been conducted in two different countries (Ghana, Senegal) and had twelve individual farmers participating in it. The table below shows the percentage of votes that every use case received following the rating (High, Medium, Low), and the partner choices.

Table 51: Agriculture uses cases Survey

| Use case | Survey Results | Partner's choice | Rank |
|---|---|---|---|
| **UC1 Monitor soil** | High - 77.78%<br>Medium - 11.11%<br>Low - 11.11% | 2 | 2 |
| **UC2 Monitor Field Weather Data** | High - 88.89%<br>Medium - 11.11% | 1 | 1 |
| **UC3 Storage Moisture and Temperature** | High - 77.78%<br>Medium - 11.11%<br>Low - 11.11% | 3 | 3 |

## 4.2 Fish Farming

The fish farmer use cases survey has mostly be conducted in Ghana and had eighteen individual farmers participating in it. The table below shows the percentage of votes that every use case received following the rating (High, Medium, Low), and the partner choices.

Table 52: Fish Farming uses cases survey

| Use case | Survey Results | Partner's choice | Rank |
|---|---|---|---|
| **UC1 Fish Pond Water Quality** | High -94.44%<br>Medium – 5.56% | **1** | 1 |
| **UC2 Estimate Fish Quantity** | High – 83.33%<br>Medium – 16.67% | 2 | 2 |

## 4.3 Water

The water use cases survey has been conducted in two different countries (Ghana, Senegal) and had fifteen individual end users participating in it. The table below shows the percentage of votes that each use case received following the rating (High, Medium, Low), and the partner choices.

Table 53: Water uses cases survey

| Use case | Survey Results | Partner's choice | Rank |
|---|---|---|---|
| **UC1 Measure Water Level** | High - 86.67%<br>Medium- 13.33% | **-** | 3 |
| **UC2 Water quality** | High – 100% | - | 1 |
| **UC3 Prediction of the duration of water reserve** | High – 100% | **-** | 1 |

## 4.4 Urban and Environment

The environment use cases survey has been conducted mostly in Ghana and had five individual end user and developer participating in it. The table below shows the percentage of votes that each use case received following the rating (High, Medium, Low), and the partner choices.

Table 54: Environment uses cases survey

| Use case | Survey Results | Partner's choice | Rank |
|---|---|---|---|
| **UC1 Early Flood Detection** | High - 60%<br>Medium- 0%<br>Low – 40% | 1 | 1 |
| **UC2 Monitor Air Quality** | High - 20%<br>Medium - 20%<br>Low – 60% | **2** | **2** |
| **UC4 Monitor Septic Tank Level** | High - 20%<br>Medium- 20%<br>Low – 60% | **3** | 3 |

## 4.5 Logistics & Transports

The logistics and Transports use cases Survey has been conducted in Ghana and had five individual end users and workers participating in it. The table below shows the percentage of votes that each use case received following the rating (High, Medium, Low), and the partner choices.

Table 55: Logistics & transport uses cases survey

| Use case | Survey Results | Partner's choice | Rank |
|---|---|---|---|
| **UC1 Track, operations, and remote monitoring** | High -60%<br>Medium- 20% | **2** | 2 |

| | | | |
|---|---|---|---|
| | Low – 20% | | |
| **UC2 Real-time visibility across the supply chain** | High -40%<br>Medium- 60% | 1 | 1 |
| **UC3 Check the integrity, identification, authentication and traceability of goods** | High -20%<br>Medium - 60%<br>Low – 20% | **3** | **2** |

## 4.6 Summary of the users feedback

In summary, the user survey shows that the most relevant/popular use cases are:

- **Agriculture: UC2 Monitor Field Weather Data**
- **Fish farming: UC1 Fish Pond Water Quality**
- **Water: UC2 Water quality and UC3 Prediction of the duration of water reserve**
- **Environment: UC1 Early Flood Detection**
- **Logistics and transports: UC2 Real-time visibility across the supply chain**

This survey result will help us build the pilots real-life validation of use cases.

# 5 PILOTS FOR REAL-LIFE VALIDATION

In this chapter we present a detailed analysis of the different pilot sites selected for WAZIUP project, including a description of each facility with pictures and the link to the corresponding use cases. The five pilots are precision agriculture, cattle rustling, logistic & transport, fish farming and urban waste management.

## 5.1 Precision agriculture

The goal of this pilot is to improve the working conditions and yield in the agricultural field by giving precise information on the ground status. To achieve this, we will gather data on the environmental conditions with dedicated sensors, analyze data and make optimized and personalized predictions for the farmers. Table 56 gives the leaders of the pilot.

**Table 56: Precision agriculture pilot leaders**

| | |
|---|---|
| Pilot Leader | UNPARAL |
| Hardware Leader | UNPARAL/UGB |
| Software Leader | CNET |
| Application Leader | FL |

Two experiments will be conducted: one in Ghana and another in Senegal, involving smallholder farmers and a university in an operational farm environment. This configuration will give the opportunity of a strong interaction between farmers, professionals and students and thus will give bigger chance for the pilot to succeed.

The first pilot site is located in UGB demonstrational farm in Senegal. The deployment field is 35x20m wide and linked to an irrigation system. Different types of crops are cultivated in the farm such as tomatoes and onions. The second pilot site consists of several small farms located in Ashanti region of Ghana, in Fomena, Tepa, Nyinahin and Gyinase. The farmers cultivate vegetables, maize, cassava and cocoa. The size of the individual farms for maize, cassava and cocoa cultivation has a minimum area of 1 acre (4046.86 m2) to 2 acre (8093.71m2). The vegetables are cultivated by one farmer on a minimum of 52 beds of a size 2x10m each. Figure 10, Figure 11 and Figure 12 shows pictures from the terrain.

**Figure 10: UGB Farm with Professional and Students in Senegal**

Figure 11: A vegetable bed and a cocoa farm in Ghana

Figure 12: Smallholder vegetable Farms in Ghana

### 5.1.1 Validation Cases

The following section lists the use cases that will be validated in the pilot sites.

**UC1: Monitor soil**:

- Monitoring of farm soil moisture and nutrients: potassium, nitrogen, etc.
- Algorithm based on photographic data analyses of pest count.

**UC2: Monitor Field Weather Data:**

- This use case will be tested in various farms
- Real-time monitoring the field weather situation (temperature, precipitation, humidity, wind speed, etc.)
- Algorithm based on historic data analytic for weather prediction.

**UC3: Storage Moisture and Temperature**:

- Real-time monitoring of storage conditions.
- Monitoring of commodity moisture condition in storage
- Real-time notification if storage condition goes out of the ranges set

Pest situation on the farm:

- Periodic monitoring of pest situation on the farm

- Algorithm based on data analytic of soil composition
- Ability to get notification of what to do/apply on the farm

### 5.1.2 Infrastructure setup

The main needs for this use case validation will be the IoT nodes responsible for sensing the corresponding parameters. A number, 10 to 20, of hardware motes will be deployed in the various pilots field. These devices will be deployed for capturing temperature, humidity, pH data from the sensors. Gathered data will be routed to the gateway node, which stores the data in them locally and/or sent it to other machines (central servers), through the different interfaces provided by it.

Taking into account experimentation and service provision prosecuted by the project, there will be the need to define an infrastructure that allows executing both experiments and user-addressed services concurrently, thus providing flexibility for researchers to try their applications on the test bed, at the same time that an end user service will be running.

## 5.2 Cattle rustling, Saint-Louis, Senegal

Cattle rustling is a serious problem observed in African countries, particularly in Senegal. This is a recurring phenomenon that causes a lot of problems to farmers. Cattle's stealing is extremely expensive; it represents millions for farmers but also for the state annually. Faced with this problem, the famer is often helpless. Table 57 gives the leaders of the pilot.

**Table 57: Cattle rustling pilot leaders**

| | |
|---|---|
| Pilot Leader | UPPA |
| Hardware Leader | UPPA |
| Software Leader | CNET |
| Application Leader | C4A |

With the collaboration of the surrounding famers, the goal of the pilot is to put sensors around the cows' neck in order to have measure related to location and speed. Collaboration with CIMEL (Centre d'Impulsion et de Modernisation de l'Elevage), which is a public structure for livestock impulsion and modernization, is planned, to help with the tests and a cattle rustling case simulations. Figure 13 and Figure 14 shows pictures from the terrain.

Figure 13: Farm in a rural area around UGB

Figure 14: CIMEL Center (cattle feeding)

### 5.2.1 Validation Cases

The following section lists the uses cases that will be validated in the pilot sites.

**UC1: Real-time position and itinerary of the cattle's herd**:

- Ability to get real-time position of the cattle's herd
- Ability to trace route of the cattle's

**UC2: Ability to receive notification in critical situations**

### 5.2.2 Infrastructure setup

The use of technologies could be a solution for this problem, by ensuring prevention but also to recover the stolen cattle. Indeed, the use of sensors can enable us to trace the cattle by collecting information regularly on the position and the speed of the movement of animals. Selected sensors can be combined with collars that will be placed around the neck of the animal, although it may be less effective because thieves can remove the clamps. For added security we can consider placing subcutaneous sensors.

We intend to deploy in the first time hardware motes within the deployment field. These devices will be deployed for capturing real-time position, movement, velocity, etc. data from sensors. Gathered data are routed to the gateway node, which stores the data in them locally and/or sent to other machines (central servers), through the different interfaces provided by it.

## 5.3 Logistics and transport, Saint-Louis, Senegal

Logistic and Transport are fundamentally about moving things from one place to another. Therefore, the main service components of Logistic and Transport can be categorized into the things that move and the things that do the moving—the "demand" and "supply" sides of logistics. Whether by air, ground or sea, transportation and logistics are essential components to many enterprises' productivity, and access to real-time data is critical. Many industries and business sectors are struggling to grasp the possibilities of data-driven technology, but companies in transport and logistics are way ahead. By their very nature, the logistics providers that move objects by air, sea, rail, and ground have widely distributed networks and rely on rapid information about those networks to make decisions. As a result, they were quick to see the benefits of new sensor and connection technology, placing them at the forefront of the transition to a connected world. Table 58 gives the leaders of the pilot.

**Table 58: Logistics and transport pilot leaders**

| | |
|---|---|
| Pilot Leader | CNET |
| Hardware Leader | UPPA |
| Software Leader | CNET |
| Application Leader | C4A |

We will work with the following companies established in Senegal in order to improve their conditions of work and transport of goods:

- Sogefret worked in Freight / Cold Storage / Perishable
- Transport Frigorifique Terrestre - Senegal SARL - TFT-SENEGAL

Figure 15 show pictures from the terrain.


Figure 15: In the first caption, trucks transporting freeze foods.

### 5.3.1 Validation Cases

This section lists the use cases that will be validated in the pilot sites.

- **UC1: Track operations and remote monitoring.**
- **UC2: Real-time visibility across the supply chain.**
- **UC3: Check the integrity, identification, authentication and traceability of goods.**

### 5.3.2 Infrastructure setup

The use of technologies could be a solution for this problem. The use of sensors can enable us to track objects by collecting regularly information on the position of the movement of vehicles. The infrastructure set up with this application comprises the following steps: first, collaboration is settled with the structure in order to deploy the application. Then, a test bed will be deployed in concordance with the resources that the structure proposes. Sensors and required devices will be deployed for capturing geographical position and speed data.

## 5.4 Fish farming, Kumasi, Ghana

In order to increase the management efficiency of the fish farms, this pilot will deploy a network of sensors to monitor remotely and in real time the water situation and quality within the fish ponds. The Table 59 gives the fish farming pilot leaders.

**Table 59: Fish farming pilot leaders**

| | |
|---|---|
| Pilot Leader | EGM |
| Hardware Leader | UPPA |
| Software Leader | CNET |
| Application Leader | FL |

Two different test bed scenarios will be used to effectively integrate the real world fish farming situations/solutions with the digital world of WAZIUP. The first test-bed will be hosted by Lazarus Farms in Ekyem, Fumesua- Kumasi. Lazarus Farms is a semi-intensive fish farming business operated by two farm hands. They own a fish farm of 4 mini ponds of 90x90m. The second test-bed will be hosted by Kumah Farms[1] in Domeabra, Odoum- Kumasi. It is a large farm with 22 ponds of different sizes. Kumah Farms have an eco-tourism, which undertakes activities such as crop farming, livestock farming, apiculture and aquaculture. Those two farms are raising both tilapia and catfish. Figure 16 and Figure 17 gives pictures from the terrain.

**Figure 16: Fish farming pond (Small size fish pond on Lazarus and Kumah Farms)**

**Figure 17: Fish farming pond (Different Sizes of large fish ponds in Kumah Farms)**

[1] http://www.kumahfarms.com

### 5.4.1 Validation Cases

The following section lists the uses cases that will be validated in the pilot sites.

**UC1: Fish Pond Water Quality:**

- Real-time monitoring the water quality, dissolved oxygen, water temperature and water level

**UC3: Cost-Efficient Feeding System:**

- Algorithm based on data analytic for the recommendation of fish feeding

### 5.4.2 Infrastructure setup

The main needs for this use case validation will be the IoT nodes responsible for sensing the corresponding parameter (temperature, CO, etc.) and communicating with the Gateways which stores the data in them locally and/or sent to other machines (central servers), through the different interfaces provided by it (WiFi or GPRS/UMTS). Taking into account, experimentation and service provision, prosecuted by the project, there is need to define an infrastructure that allows executing both experiments and user-addressed services concurrently, thus providing flexibility for researchers to try their applications on the testbed, at the same time that an end user service is running.

## 5.5 Environment and urban agriculture

African cities have the fastest urbanization speed of the world. Some cities like Kinshasa will have its population tripled by 2050. Thus, the African urbanity becomes the perfect experimental field to test urban smart systems. The most important challenges are the household living conditions improvement of food security, appropriate waste management and digitalization of the different sectors. Table 60 gives the pilot leaders.

**Table 60: Urban and domestic pilot leaders**

| | |
|---|---|
| Pilot Leader | IT21 |
| Hardware Leader | UPPA/WOELAB |
| Software Leader | CNET |
| Application Leader | PUDB |

WAZIUP pilot cases in Togo will be incorporated with the Smart city project HubCité and this project will constitute the test-bed. WAZIUP will deploy two testing sites in Lome i) pilot case urban waste management and ii) pilot cases urban organic food. These two pilot cases will be tested with the following startups incubated in WoeLab to improve their conditions of work and help them develop new innovation solutions.

- SCoPE[2]
- Urbanattic

[2] https://www.youtube.com/watch?v=J5qL66I9xp8

On urban waste management, the pilot will connect the real world network of plastic waste collection kits, deployed in households by SCOPE (Sorting and Collection of Plastics in our environment) startup. SCOPE is a startup that wants to connect effectively, thanks to a web device / mobile adapted, the problem of plastic waste recycling solutions by collecting directly from households. The kit used so far is the low-tech (no intelligence) and users must send a signal themselves by phone when they see the plastic waste box is filled. Within WAZIUP, we want to automate the process, making the bin smarter. We will be testing the "smart" kits made by the startup so that they are able to notify once the waste bin is full.

In this test-scenario another model will also be validated, the startup Woebots/SCOPE wants to test a dynamic process of collecting electronic waste on the model of itinerant buyers "gakpoglegble". These collectors will be equipped with GPS devices that allow all users who have subscribed to the program to recycle e-waste geotag and contact them when they are in their environment.

On urban organic food, Urbanattic have different cultivation scattered around the city and cannot monitor all at once. Urbanattic need to have more direct feedback on reliability and performance. This should overcome the human hazard (stronger in the town production). Through the platform, the startup has to monitor the output and organize team ace hand of an exploded activity. There is some pollution issues in urban agriculture: specific agricultural threats to the city like air quality, heavy metals in soil, moisture proof stress measurement, vegetable pesticides, etc. Every site of generation or storage will be connected to the platform via sensors collecting light, humidity and temperature information. Figure 18 and Figure 19 shows pictures from the terrain.

**Figure 18: Urban waste management in Lomé with SCoPE**

Figure 19: Urban organic food

### 5.5.1 Validation Cases

The following use case lists what will be validated in the pilot sites.

**UC1: Indoor/small scale farming automation**:

- Real-time monitoring the vegetable garden
- Real-time monitoring the environment (soli and air quality), heavy metals in soil, moisture proof stress measurement, vegetable pesticides, etc.
- Rea-time monitoring light, humidity and temperature of the FoodLab, warehouse

**UC3: Confirm emptied waste bins**:

- Check the integrity, identification, authentication and traceability of waste disposal dispositive

- Get notification from waste bit once it is full
- Track and geo localization of waster collector
- Automate the waste collection process

### 5.5.2 Infrastructure

In this section the infrastructure setup of both pilot cases are described. For urban waste management, the test should be done on the currently deployed twenty bins kits. The equipment in use: temperature and humidity sensor, GPS module to geotag the *Arduino Uno* kit, chargeable battery 9V, Arduino connection (male-male, male-female, female-female) Data collected in a perimeter with a radius of 1km thanks to an IOT node, is sent to a gateway node relay installed in each plastic bank. Locally Stored data are then transferred to the central server of the startup (For now, there is only one local point of existing central sorting with 22 users in its perimeter). The infrastructure will allow the implementation and test other possibilities related applications launched by developers

For urban organic food, the infrastructure and logistics need will be the same as the test bed for plastic collection project; since Urbanattic is also organized in a network of storage locations, each for a perimeter with a radius of 1km. Each granary of this network could host the IOT note responsible for sensing the parameters (capturing temperature, humidity, pH data, etc. from sensors) and be equipped with the relay gateway to store data.

## 5.6 Summary of the pilots

Five pilot deployments have been created out of the 31 use cases. Table 61 is summarizing the uses cases covered by each pilots.

**Table 61: Pilots use case coverage**

| Pilot name | Use cases |
|---|---|
| **Precision agriculture** | • UC1: Monitor soil<br>• UC2: Field Weather Situation<br>• UC3: Storage Moisture and Temperature |
| **Cattle rustling** | • UC1: Real-time position and itinerary of the cattle's herd<br>• UC2: Ability to receive notification in critical situations |
| **Logistics and transportation** | • UC1: Track operations and remote monitoring<br>• UC2: Real-time visibility across the supply chain<br>• UC3: Check the integrity, identification, authentication and traceability of goods |
| **Fish farming** | • UC1: Fish Pond Water Quality<br>• UC3: Cost-Efficient Feeding System |
| **Environment and Urban agriculture** | • UC1: Indoor/small scale farming automation<br>• UC3: Confirm emptied waste bins |

From the above pilot cases, the minimal set features (one UC per domain) will be implemented following the « Minimum Viable Product (MVP) » approach within the exploitation activities of the project. In the next, under the WP4, the further analysis will be done to define the UC that will be implemented as a MVP. They are ultimately intended to create sustainable communities and new viable businesses.

**Table 62 : MVP allocation table**

| Pilot name | MVP leader (WP4) | HW incl Lora network (WP2) | Software Platform (WP3) | apps (WP3) |
|---|---|---|---|---|
| **Precision agriculture** | ULL | ULL/UGB | CNET | FL |
| **Cattle rustling** | UPPA | UPPA | CNET | C4A |
| **Logistics and transportation** | CNET | UPPA | CNET | C4A |
| **Fish farming** | EGM | UPPA | CNET | FL |
| **Urban Environment** | IT21 | UPPA/WOELAB | CNET | PUBD |

# 6 CONCLUSION

Africa is at a pivotal moment in its technological revolution and IoT and Big Data bring enormous potential of innovations. Achieving sustainable development in Africa can be done by improving the use of technology in some key domains: Agriculture, Water, Fish farming, Health, Environment, Logistics and transport, Urban and domestic and cattle to name a few.

A total of 31 use cases have been identified and analysed in those different domains. Data has been collected from field reviews and from different African partners including Startup and SMEs. Finally, a ranking has been done to help choose the use cases that will be implemented in our WAZIUP test beds. From the initial list of 31 use cases, five pilot deployments have been selected. The pilots are geared toward specific African problems: for example, the cattle's stealing is a huge problem for animal owners. Fish farming is also an economical sector with a great growth potential. Our interviews also revealed that weather predictions are of tremendous importance for farmers in Africa, as well as water management. These pilots will be implemented following the « Minimum Viable Product (MVP) » approach within the exploitation activities of the project. They are ultimately intended to create sustainable communities and new viable businesses.

# PART II: Architecture requirements and specification

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# DEFINITIONS

| Term | Definition |
|---|---|
| **Big Data** | Big Data is a domain covering the techniques used for processing large amounts of data. |
| **Device** | Devices are only technical artifacts meant to provide an interface between the digital and the physical worlds, i.e. a link between the Virtual Entities and the Physical Entities. There are three basic types :<br>• Sensor<br>• Tag<br>• Actuator |
| **Gateway** | A Gateway allows several Sensor nodes to be connected. It performs data transformation and forwarding into the Cloud. It is sometime able to run local automations, including containerized applications and database services. |
| **IoT** | The Internet of things (IoT), in WAZIUP, is a technical and business domain involving sensors collecting measurements which are exploited using the Cloud resources. |
| **Local Cloud** | A local Cloud is an infrastructure able to deliver services to clients in a limited geographical area. A local Cloud can be limited to a Gateway or a local network. |
| **PaaS** | Platform as a Service (PaaS) is a category of the cloud computing service that provides a platform allowing customers to develop, run, and manage applications. |
| **Resource** | Resources are software components that provide some functionality. Resources can either run on a Device – hence called On-Device Resources – or they can run somewhere in the network (Network Resources). |
| **Sensor** | A sensor is a special Device that perceives certain characteristics of the real world and transfers them into a digital representation. |
| **Sensor node** | A Sensor node is the physical platform where one or several sensors are connected. It transmits the data measurements to the Gateway. |
| **Service** | • Resource-level Services: Expose the functionality, usually of a Device, by accessing its hosted Resources.<br>• Virtual Entity-level Services: Provide access to information at a Virtual Entity-level.<br>• Integrated Services: Service composition of Resource-level or Virtual Entity-level Services as well as any combinations of both Service abstractions. |
| **Virtual Entity** | Representation of Physical Entities in the digital world. |

# 1 INTRODUCTION

In order to bridge the gap between rural Africa need and the current IoT offer, the WAZIUP project aims at developing an IoT/Big Data platform that corresponds to users needs. Based on the use needs and use case scenarios presented in the first section, we elaborate the full architecture of the WAZIUP platform. First of all, several new concepts are proposed to respond to those user needs: the Platform as a Service (PaaS) approach to IoT, the data processing capacity inspired from Big Data techniques and finally the local and global Cloud. The idea of extending the PaaS approach to IoT is to propose a platform dedicated to IoT developers that can reduce the time-to-market for an application by cutting the development costs. The Big Data techniques enable the processing of the huge amount of data produced by sensors. Those techniques allow creating actionable information and knowledge out of the raw data. Finally the local and global Clouds address the intermittent connection challenge: when Internet is not available, the user can still access some IoT functionalities from the local Cloud.

This part is structured as follows: Section 2 gives an overview of the concepts and objectives that are guiding the development. Section 3 gives the architecture of the platform, including a functional overview, the actors definitions, the components and finally the sequence diagrams. Section 4 details all the requirements for each of the components of the platform. The interfaces between the components are given in Section 5. Section 6 describes the hardware platforms and networks. The data models and formats used by the platform are given in Section 7. Section 8 provides a preliminary design for each of the components. Section 9 shows a state of the art, both on IoT and Big Data. Finally the deliverable is concluded in Section 10.

# 2 CONCEPT AND OBJECTIVES

This section sums up the concepts that underpin the WAZIUP platform, which are three: the PaaS approach to IoT, the data processing capacity inspired from Big Data techniques and finally the local and global Cloud. This section will also reveal the objectives of the platform.

## 2.1 PaaS approach

Platform as a Service (PaaS) is a category of the cloud computing service that provides a platform allowing customers to develop, run, and manage applications; without the complexity of building the infrastructure typically associated with developing and deploying applications. Typically, a PaaS framework will compile an application from its source code, and then deploy it inside lightweight virtual machines or containers. This compilation and deployment is done with the help of a file called the manifest, which allows the developer to describe the configuration and resource needs for the application. The manifest file will also describe the services that the application requires and that the platform will need for provision. Furthermore, PaaS environments usually offer an interface for applications to scale up or down, or to schedule various tasks within the applications.

The idea of WAZIUP is to extend the paradigm of the PaaS to IoT. Developing an IoT Big Data application is a complex task. A lot of services need to be installed and configured, such as databases and complex event processing engines. Furthermore, it requires an advanced knowledge of the various communication protocols, the programming of embedded devices, the storage, processing and analysis of the data in a distributed fashion and finally the programming of GUIs and user interactions. The promise of the PaaS extended to IoT is to abstract away this work to a large extent.

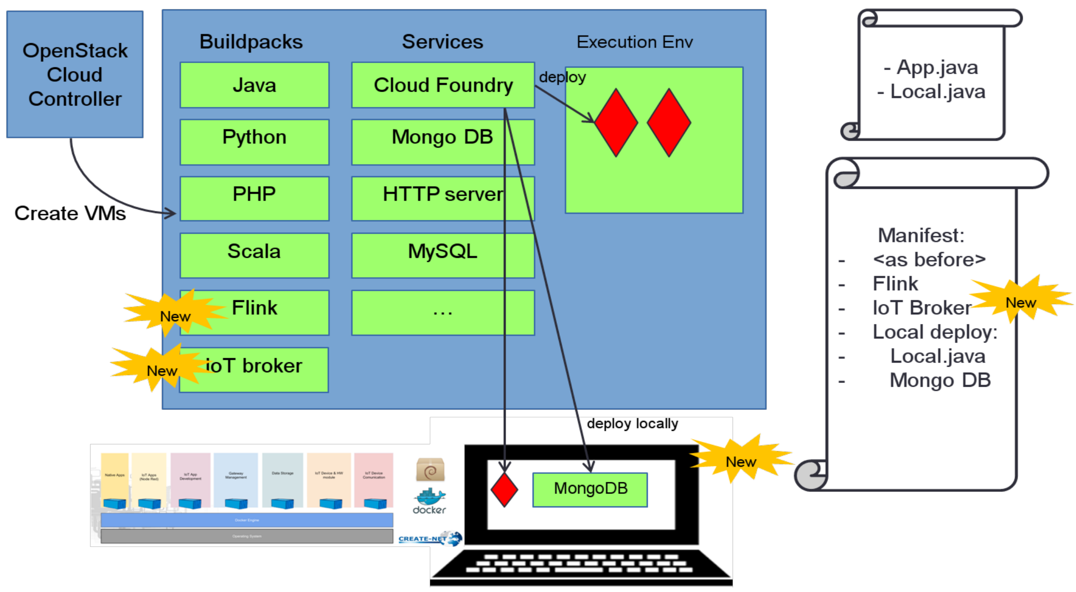


Figure 20: PaaS deployment extended for IoT in WAZIUP

Figure 20 shows the PaaS deployment in WAZIUP. Traditional PaaS environments are usually installed on top of IaaS (in blue in the picture). The blue boxes are physical servers, respectively the Cloud Controller and one Compute node. The PaaS environment is then installed inside the IaaS VMs, in green in the picture. We use Cloud Foundry as a PaaS framework. It comes with a certain number of build packs, which and programming languages compilers and runtime environments. It also provides a certain number of preinstalled services such as MongoDB or Apache Tomcat. The manifest file, showed on the right hand side, provides a high-level language that allows describing which services to instantiate. We propose to extend this language to IoT and big data services such as:

- Data stream and message broker
- CEP engines
- Batch processing engines
- Data visualization engines

Furthermore, we propose to include in the manifest a description of the IoT sensors that are required by the application. This query includes data such as the sensor type, location and owner. The manifest also includes the configuration of the sensors. The application will then be deployed both in the global Cloud and in the local Cloud.

## 2.2 Data Processing and Architecture

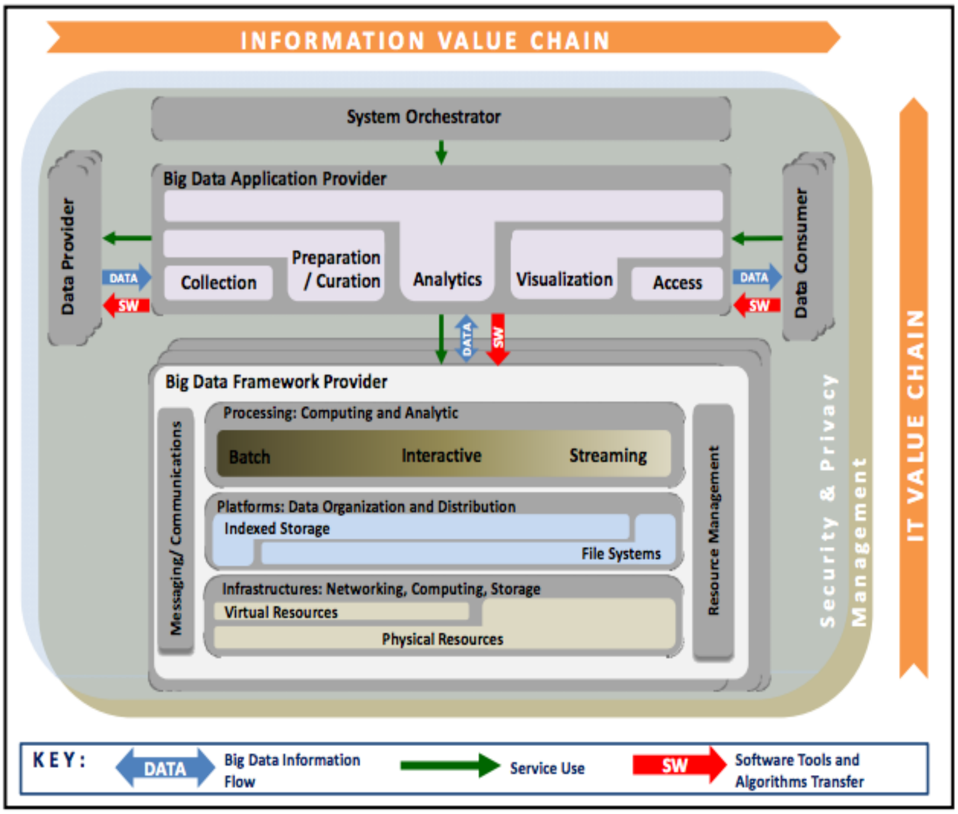


Figure 21: Big data standard architecture

Data Processing is a customary practice when data are collected in any networked and digitized system for the sake of monitoring or actuation. In particular, the recent trends in sensorization have made it possible to collect large amounts of data and to combine several data sources in order to extract the most valuable information. However, while this is indeed a core enabler for innovation, the requirements for the related architecture, including indeed the one in WAZIUP, are to potentially handle so called Big Data streams which arise from data intensive applications. Those types of data require specific technologies in order to perform task-specific analytics. In this section we refer to the output produced by the NIST Big Data Public Working Group (NBD-PWG), which is also providing documentation for basic concepts and architectural recommendation related to Big Data technologies (see Figure 21). The typical way to define Big Data applications is through the customary 3V paradigm which defines the three dimensions against which to measure the Big Data characteristics:

1. *Variety:* data can be collected by a wide set of sources, such as messages, sensors data, GPS logs from smartphones and phone traffic. For example, complementing IoT data on soil conditions with weather forecasts data may provide more accurate predictions than taking each data stream separately.
2. *Volume:* above a certain threshold, full in memory processing becomes impossible; this in turn implies several limitations in the implementation of data processing. For example, it may require the serialization of processes in order to pipeline data elaboration in parallel threads.
3. *Velocity:* for many use cases the sample rate determines the speed at which data should be processed and therefore the requirements for the processing units, real-time or nearly real-time information may represent a competitive advantage in certain markets (e.g., financial markets or smart grids).

In the following, we report on the functional blocks and the description of the actors who are part of the architecture.

- *System Orchestrator:* the typical actors which may operate the orchestration functionalities are represented by researchers and data scientists who have to ensure that the data analytics service is able to meet the specific requirements for the target service offered to customers. For instance, in the case of business models in precision agriculture, the business goal is to provide the current state of crops. The orchestration operation should ensure that the ultimate information provided to the end user is actually functional to the needs of the crops owner (e.g., it should be consistent with the production cycle and the level of maturation of crops). For instance, data analytics could offer a visualization service: the orchestration unit should ensure the correct level of resolution and the required granularity with respect to the target application (e.g., specific crop type). In order to do so, several requirements are handed over to the application provider, e.g., the entity who provides the actual data-enabled application. Among others, the main tasks of the orchestrator covers: 1) the selection of appropriate data sources, 2) collection and storage requirements and monitoring 3) the choice and adaptation of suitable analytical models (e.g., using specific models for the comfort of fisheries depending on the chemicals found in the water).
- *Data Provider:* the main task of the data provider it to perform actual data collection. This may be data captured from various sources, potentially including web apps and actual sensors. The data provider may be represented by the owner of a set of sensors deployed in a certain area or a network operator sampling regularly the connectivity logs of the mobile served by the networks owned.. The Data Provider is also responsible for performing the pre-processing of data streams, e.g., data annotation, the creation of metadata, performs appropriate signaling when new data are made available and guarantees the persistence of data.
- *Data Application Provider:* the tasks related to the data application provider are those typically of the entity in charge of developing the actual application fetching data, e.g., a sensor reading application or a mobile phone application performing periodic reporting. Hence, the data application provider is in charge of all operations performed locally in order to fetch and prepare data. This includes: 1) collection of data at the device level through

appropriate transport protocols and data formats, 2) several preparation operations for data retrieved, including partition of data, outlier removal, validation (e.g., checksums and format checks), 3) visualization of data being retrieved, 3) analytics (in-network processing), 4) ruling access to data. Very relevant to this logical role is to enforce security and privacy requirements.

- *Data Framework Provider:* it is the entity which provides the services in order to enable the Big Data applications. The typical example is the case of a cloud provider which supports the back-end of an application fed by specific data flows. In particular, it may handle several components, including networking, storage and computing. Again, the champion actor in this context is a cloud provider exposing specialized APIs to the developer of applications which are based on Big Data analytics. The Data Framework Provider is in charge of providing the physical storage, the logical storage of data and defining the policies for storage (distributed or centralized). The Data Framework Provider also decides the processing frameworks, including the processing services offered by the platform, such as video processing, temporal and streaming, etc.
- *Data Consumer:* it is the actor requiring the output of services based on Big Data. E.g, it may require visualization of data, local analysis or support of new analytics trends with respect to specific use cases. The data application provider is in charge of delivering an application to data consumers in order to perform such tasks, e.g., a visualization toolkit, or a recommender for specific actions to be taken.

## 2.3 Local and global Clouds

The WAZIUP project defines two different types of "Cloud": the local Cloud and the global Cloud. A local Cloud is an infrastructure able to deliver services to clients in a limited geographical area. The local Cloud replicates some of the features provided by the traditional Cloud. It is used for clients that may not have a good access to the traditional Cloud, or to provide additional processing power to local services. In order for such an infrastructure to be considered as a local Cloud it must support a virtualization technology. In the case of WAZIUP, the local Cloud comprises the end user or service provider PC and IoT Gateway. The local Cloud characteristics are:

- Existence of IoT devices attached
- Can have geographical characteristics
- Must support virtualization
- Must support local cloud components
- Has an identifiable administrator/owner
- Has certain regulations/privacy considerations for data access and treatment

The global Cloud, on the other end, is a "backbone infrastructure" which increases the business opportunities for service providers and allows services to access a virtually infinite amount of computing resources. In order for such an infrastructure to be considered as a global Cloud it must support a virtualization technology and be able to host the global cloud components of the WAZIUP architecture.

## 2.4 Objectives

The WAZIUP platform needs to address a number of challenges. Those challenges are related to the specific environment in which the platform will be deployed, and the needs of its end users. When deploying IoT in the context of Sub-Saharan Africa, it is necessary to target the removal of three major barriers:

1. *Lower-cost, longer-range communications*
   Vast distances and poor infrastructure isolate rural areas, leaving those who live there often poorly integrated into modern ICT ecosystems. Deploying IoT in this context must use longer range wireless communication to decrease both the complexity and the cost of data collection. Using the telco mobile communication infrastructure, when coverage is available, is still very expensive (e.g. GSM/GPRS) and definitely not energy efficient for autonomous devices that must run on battery for months. Recent so-called Low-Power Wide Area Networks (LPWAN) such as those based on Sigfox(™) or Semtech's LoRa(™) technology definitely provide a better connectivity answer for IoT as several kilometers can be achieved without relay nodes to reach a central gateway or base station. When adding the financial cost constraint and the network availability, LoRa technology, which can be privately deployed in a given area without any service subscription, has a clear advantage over Sigfox which coverage is entirely operator-managed. Some LoRa community-based initiatives such as the one promoted by TheThingNetwork[3] may provide interesting solutions and feedbacks for dense environments such as cities but under the agriculture/micro and small farm business model an even more adhoc and autonomous solution need to be investigated and deployed. On the software side, the software service platform will also need to offer highly innovative monitoring, recommendation, notification services based on the data coming from multiple rural application sectors, taking into account that, in most cases, the mobile phone is the unique technological terminal for end-users.

2. *Cost of hardware and services*
   The maturation of the IoT market is happening in many developed countries: innovative and integrated products are available for smarter home and various monitoring applications. While the cost of such devices can appear reasonable within developed countries standards, they are definitely still too expensive for very low-income sub-Saharan ones. The cost argument, along with the statement that overly integrated components are difficult to repair and/or replace definitely push for a Do-It-Yourself (DIY) and "off-the-shelves" design orientation. To be sustainable and able to reach previously mentioned rural environments, IoT initiatives in developing countries have to rely on innovative and local business models. We envision mostly medium-size companies building their own "integrated" version of IoT for micro-small scale services. In this context, it is important to have dedicated efforts to design a viable exploitation model which may lead to the creation of small-scale innovative service companies.

   The availability of low-cost, open-source hardware platforms such as Arduino is clearly an opportunity for building low-cost IoT devices from mass-market components. For instance, the Arduino Pro Mini based on an ATmega328 in its 3.3v and 8MHz version offers an excellent price/performance/consumption tradeoff. Originally designed by Sparkfun, this board can also be purchased for less than 2 euro per piece from Chinese manufacturers. It can be used to provide a platform for generic sensing IoT with LoRa long-range transmission

[3] https://www.thethingsnetwork.org/

capability. In addition to the cost argument (cost can be less than 15 euro for a fully operational long-range sensing device) such mass-market component greatly benefits from the support of a world-wide and active community of developers.

3. *Limit dependency on proprietary infrastructures, provide local interaction models* Once data are collected on the gateway, they usually have to be pushed/uploaded to some Internet/cloud servers for storage and visualization; and eventually for further processing tasks. It is important in the context of developing countries to be able to use a wide range of infrastructures and, if possible, at the lowest cost. Fortunately, along with the global IoT uptake, there is also a tremendous availability of sophisticated and public IoT clouds platforms and tools, offering an unprecedented level of diversity which contributes to limit dependency on proprietary infrastructures. Many of these platforms offer free accounts with limited features but that can already satisfy the needs of most agriculture/micro and small farm business models we are referring to when addressing IoT for Sub-Saharan African applications. What are the impacts on the design architecture/choices of the deployed IoT platforms? One simple design orientation is to highly decouple the low-level gateway functionalities from the high-level data post-processing features, privileging high-level languages for the latter stage (e.g. Python) so that customizing data management tasks can be done in a few minutes, using standard tools, simple REST API interface and available public clouds.

One additional important issue that needs to be taken into account in the context of sub-Saharan Africa is the lack or intermittent access to the Internet. Data should also be locally stored on the gateway which can be directly used as an end computer by just attaching a keyboard and a display. This solution perfectly suits low-income countries where many parts can be found in second-hand markets. The gateway should also be able to interact with the end-users' smartphone through WiFi or Bluetooth to display captured data and notify users of important events without the need of Internet access as this situation can clearly happen in very remote areas.

The high level and most important requirements for the WAZIUP Cloud platform have been grouped into the following table. They have been derived from the analysis of the use cases. The complete requirement list of the platform can be found in Section 4. The goal of this architecture document is to present a software architecture that fulfills those requirements.

**Table 63: High level requirements of the WAZIUP platform**

| Requirement | Justification |
|---|---|
| Allow users to register, identify and access selected resources | Needed to personalize the interface, and also for security and privacy reasons |
| Host the applications derived from the use cases | Cloud hosting can be a cost effective way of running an application (instead of owning hardware). |
| Provide Cloud GUIs and mobile GUIs templates. | Most Africans have mobile phones. GUI development is time consuming so providing templates make sense. |
| Support data collection/aggregation/monitoring | Local actors need to access information on the |

| | |
|---|---|
| scenarios | status of the farm/hospital. |
| Provide data analytics services | Need to extract information of higher value from raw data. |
| Support low power/low bandwidth | Many rural areas have bad internet access |
| Support offline/disconnection from internet | Many rural areas have bad internet access |
| Support SMS based applications | Many rural Africans own mobile phones (not smartphones) |
| Access to the Cloud platform should be affordable | Solution should be economical to be justified in rural context |

# 3 ARCHITECTURE

This section provides the details of the WAZIUP architecture. A functional overview is given, followed by the actor definition, the components and finally the sequence diagrams.

## 3.1 Functional Overview

This section presents the functional view of the architecture.

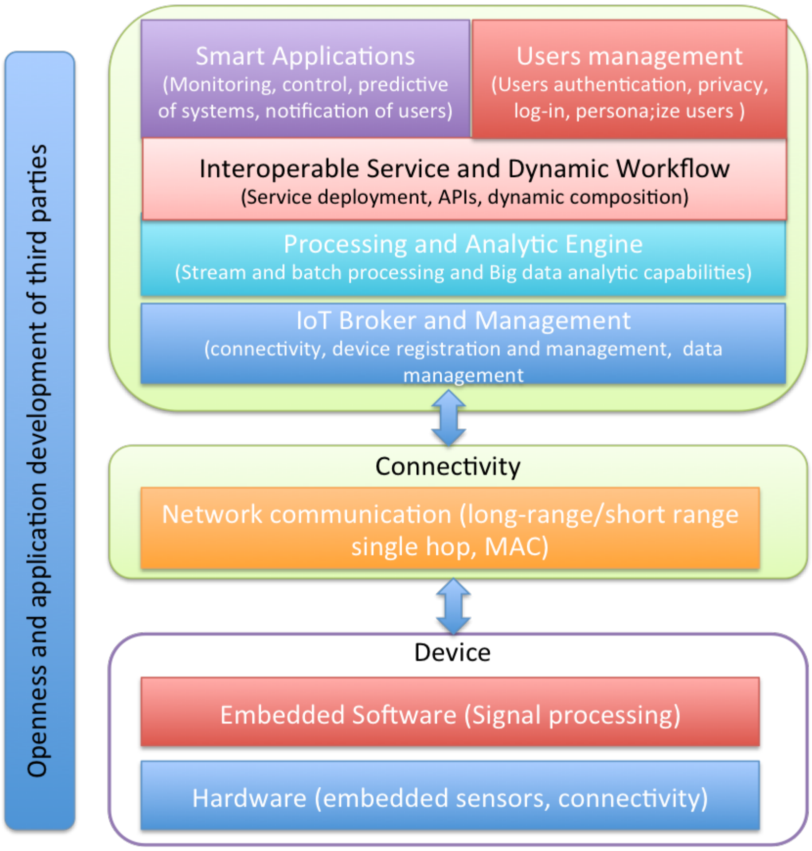


Figure 22: Functional overview of WAZIUP

Figure 22 shows the functional overview of WAZIUP. The topmost block represents the Cloud platform, the middle one is the network connectivity while the bottom one is the local deployment, including gateway and sensors. Table 64 shows the functional domains that have been identified, with a description for each of them.

Table 64: Functional domains

| Functional domains | Description |
|---|---|
| Application platform | Application writing, deploying, hosting and execution. |
| IoT platform | The connectivity of IoT devices, the sensors data and metadata. |

| | |
|---|---|
| Stream and data analytic | Data brokering, stream processing and data analytics. |
| Security and privacy | Management of the identification, roles and connections of users. Also includes data anonymisation of the data and securisation of the transmissions. |
| Platform Management | Status of the components, deployment of the platform |

## 3.2 Actors

This section gives an overview of the different actors interacting with the WAZIUP platform.

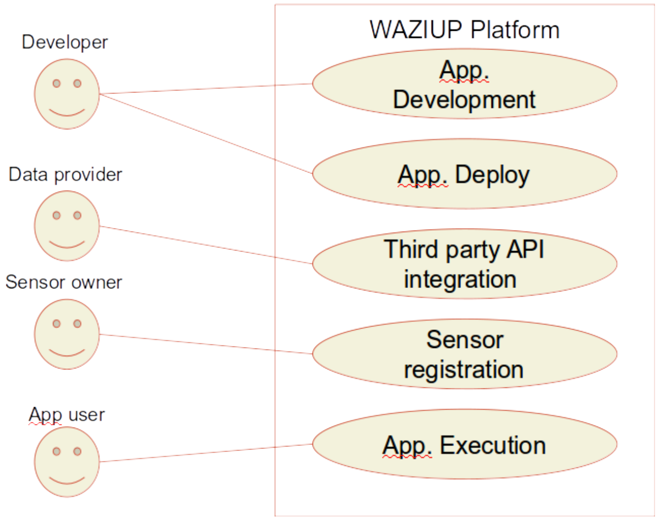


**Figure 23: Actors view of WAZIUP platform**

Figure 23 presents the actors participating in the WAZIUP platform. An actor is either a physical person or an external system. There are four actors: the developer, the data provider, the sensor owner and finally the application user. Table 65 shows the role of each actor participating in the WAZIUP platform.

Table 65: Role of the actors

| Actor | Description |
|---|---|
| Developer | The developer uses WAZIUP platform to compile and deploy his application. The application is then hosted on the platform and accessible. Their tasks are:<br>● Collect requirements for WAZIUP app<br>● Design app architecture<br>● Realize app<br>● Interact with WAZIUP API<br>● Deploy App in WAZIUP PF<br>● Maintain app |
| Sensor owner/data owner | The sensor owner deploys the sensors in the field, and then registering them on the WAZIUP platform. Sensor data then becomes available to applications. Their tasks are:<br>● Analyze deployment<br>● Install sensors<br>● Register sensors with WAZIUP<br>● Configure sensors |
| Data provider | The data provider is a third party owning an internet API to which WAZIUP is connecting in order to retrieve data. Their tasks are:<br>● Integrate third party API<br>● Access control of API<br>● Maintain the API |
| Application user | The application user is accessing the application developed by the Developer and deployed on WAZIUP. Their tasks are:<br>● Register with WAZIUP<br>● Access/use application<br>● Provide feedback on app |
| Administrator | The administrator manages and configures the platform; and administrates the users. Their tasks are:<br>● Manage users accounts<br>● Configure the platform<br>● Control resources usage |

## 3.3 Components

Figure 24 presents the full WAZIUP architecture. It shows the four functional domains: Application Platform, IoT Platform, Security and Privacy and finally Stream & Data Analytic. The Application Platform involves the development of the application itself and its deployment in the Cloud and in the Gateway. A rapid application development (RAD) tool can be used, such as Node-Red. The user provides the source code of the application, together with the manifest. As a reminder, the manifest describes the requirements of the application in terms of:

- computation requirements (i.e RAM, CPU, disk)
- references to data sources (i.e. sensors, internet - sources...)
- big data processing engines (i.e Flink, Hadoop...)
- configuration of sensors (i.e. sampling rate)
- local and global application deployment

The application source code, together with the manifest, is pushed to the WAZIUP Cloud platform by the user. The orchestrator component will read the manifest and trigger the compilation of the application. It will then deploy the application in the Cloud Execution Environment. It will also instantiate the services needed by the application, as described in the manifest. The last task of the orchestrator is to request the sensor and data sources connections from the IoT components of the architecture. The sensor discovery module will be in charge of retrieving a list of sensors that matches the manifest description. On the left side of the diagram, the sensor owners can register their sensors with the platform. External data sources such as Internet APIs can also be connected directly to the data broker. The sensors selected for each application will deliver their data to the data broker, through the IoT bridge and pre-processor. This last component is in charge of managing the connection and configuration of the sensors. Furthermore, it will contain the routines for pre-processing the data, such as cleaning, extrapolating, aggregating and averaging. Historical data can be stored using the Storage manager.

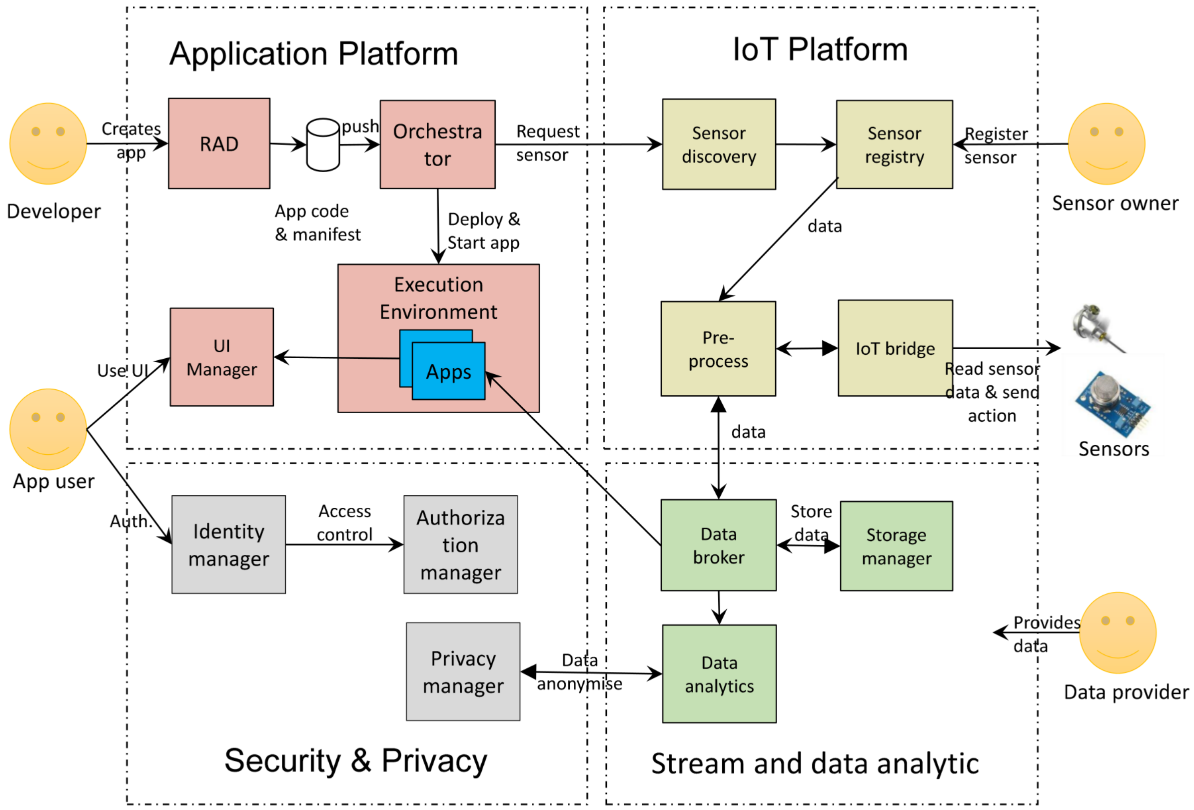


**Figure 24: Components of the WAZIUP platform**

The Security and Privacy domain contains three components: the Identity Manager, the Authorization Manager and the Privacy Manager. The first one is in charge of providing the identification, the roles and the connections of the users. The Authorization Manager provides the access policy for each of the WAZIUP resources. Finally the Privacy Manager provides services for the privacy of communication and also the anonymization of data.

The components belonging to the architecture described in this deliverable are described in Table 66.

**Table 66: Components**

| Functional domain | Component name | Description |
|---|---|---|
| Application platform | Orchestrator | The orchestrator role is to:<br>• read the manifest<br>• compile the applications<br>• deploy the applications<br>• instantiate the services needed by the applications<br>• request the sensor and data sources connections |
| | Execution environment | The execution environment allows executing the applications, based on their characteristics (language, config etc.) |
| | RAD | The RAD component provides a graphical interface allowing to develop an application (i.e. produce a source code). |
| | UI manager | The GUI manager is in charge of spawning the GUIs of the applications. |
| IoT platform | IoT bridge | The role of the IoT bridge is to ensure the connectivity between the sensors and the Platform. It is able to receive the sensor raw data using the sensor protocol. It then sends them to the pre-process module to be processed. |
| | Pre-Process | The Pre-Process component processes the received data (transform its format, apply some operator, etc) |
| | Sensor registry | The sensor registry role is to store the virtual representation of the real sensor. |
| | Sensor discovery | The sensor discovery role is to find the list of available sensors that matches a query string. |
| Stream and data analytic | Data broker | The data broker (or IoT broker) is the central component of the IoT platform. It manages information about the context. It is able to receive and provides context information. It also provides a messaging system based on subscription. |
| | Storage manager | The storage manager stores the data from the platform itself (user names, app config etc.) but also the data generated by the sensors. |
| | Data analytic | The data analytic component provides data analytic services to the applications. |
| Security and privacy | Identity manager | The identity manager provides the identification, roles and connections of users. |

| | | |
|---|---|---|
| | Authorization manager | The authorization manager provides access policy the WAZIUP resources. |
| | Privacy manager | The privacy manager provides services for privacy of communication and anonymization of data. |
| Platform Management | Status manager | The status manager controls the status of each component and of the whole platform. |

## 3.4 Sequence diagrams

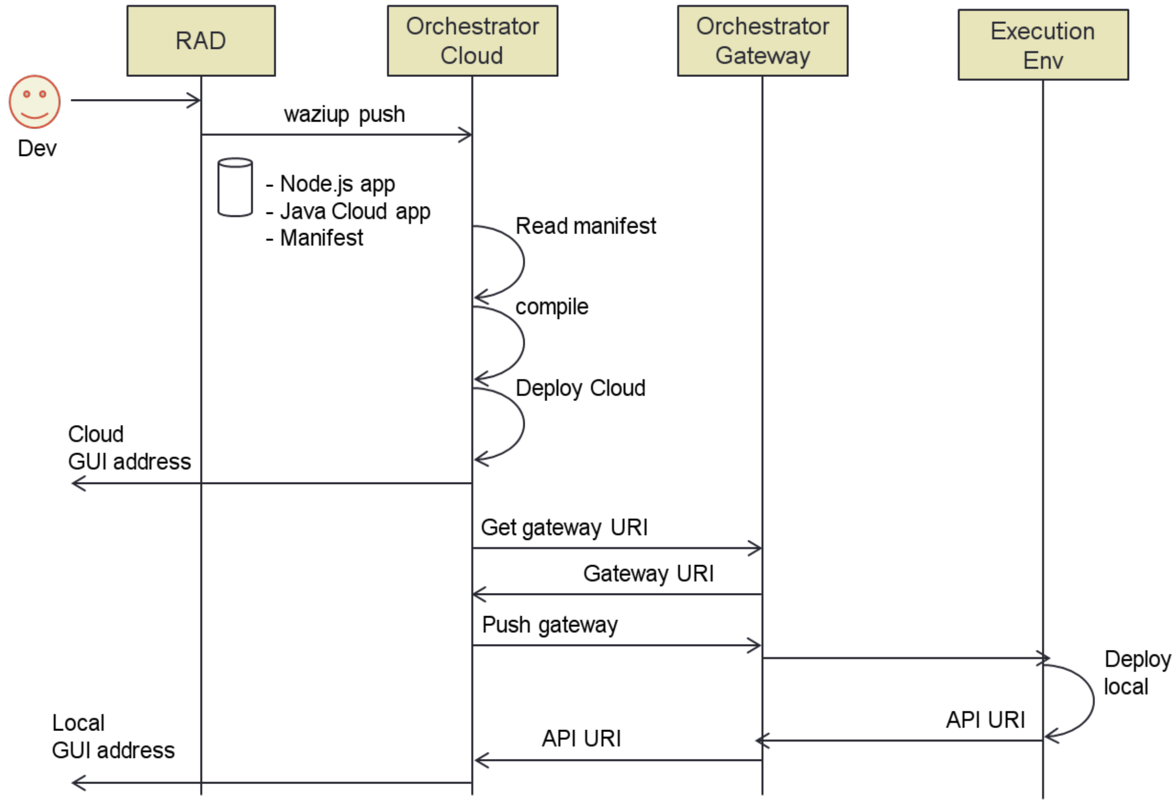


Figure 25: Application deployment sequence diagram

Figure 25 shows the sequence of an application deployment on WAZIUP platform. First, the developer uses the RAD to create a program source code. It is then pushed on WAZIUP platform. The Orchestrator of the Cloud instance reads the manifest, compiles the application and deploys it in the Cloud execution environment. Using the information from the manifest, it will also select the gateways where the application needs to be deployed. It will then contact the orchestrator from the corresponding gateways and ask them to deploy the application. The remote gateway will deploy it in its Execution Environment. If the gateway is not available at the moment, this action is delayed.

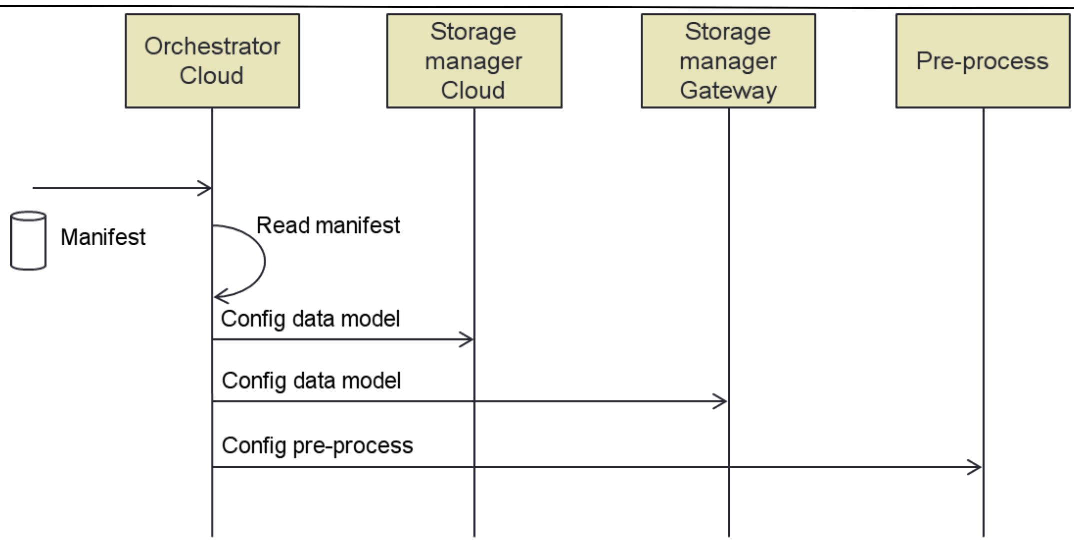


**Figure 26: Application configuration sequence diagram**

Figure 26 shows the sequence of the configuration of a new application. The Cloud Orchestrator first reads the manifest and extracts the application configuration. It then pushes that configuration to the Cloud Storage manager and to the corresponding Storage managers from the gateways. This includes the configuration of the data models in the databases. Finally the configuration is also pushed to the Pre-process component of the gateways, to allow the application to fine tune its data processing at gateway level.

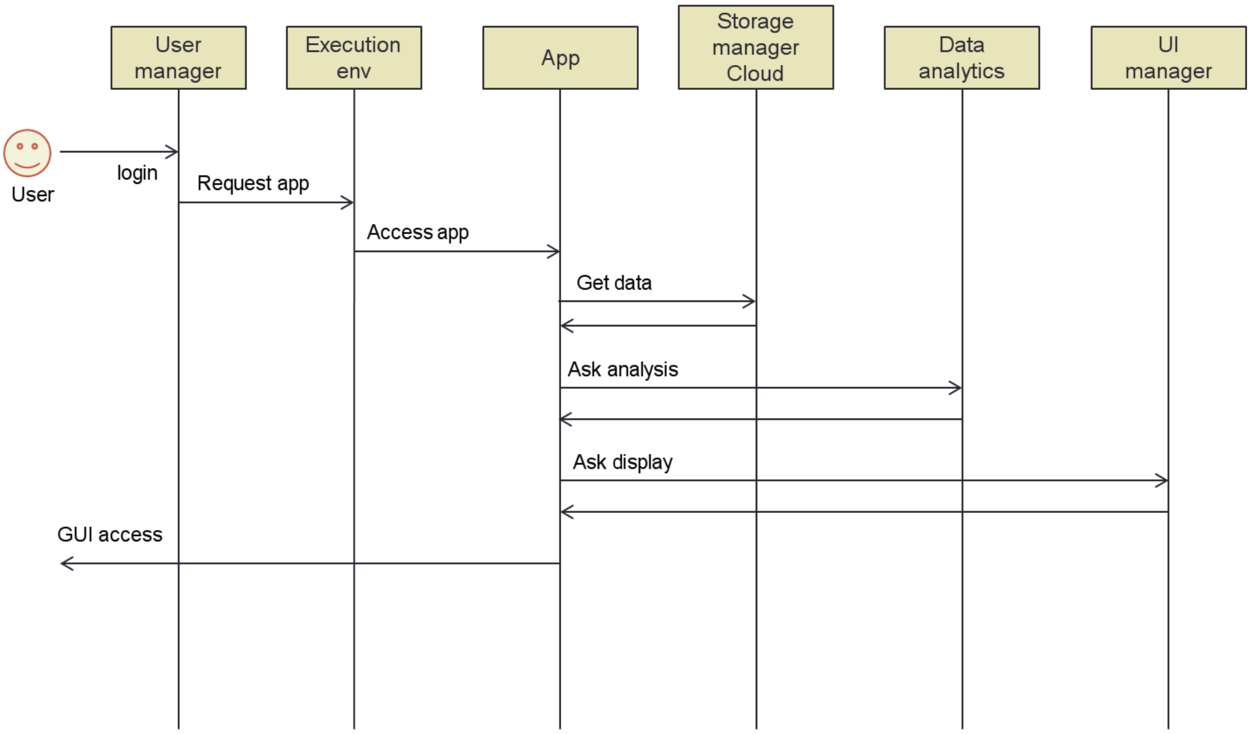


**Figure 27: Application run sequence diagram**

Figure 27 shows the sequence followed by a running application. First of all a user requests the access to an application to the user manager, using his login and password. If granted, the User manager will forward the request to the execution environment which gives access to the app. The app is then getting its data from the Cloud Storage manager. It is further pushing the data to the data analytics component in order to get the elaborated data. These data are then sent to the UI manager, in order to be displayed to the user.

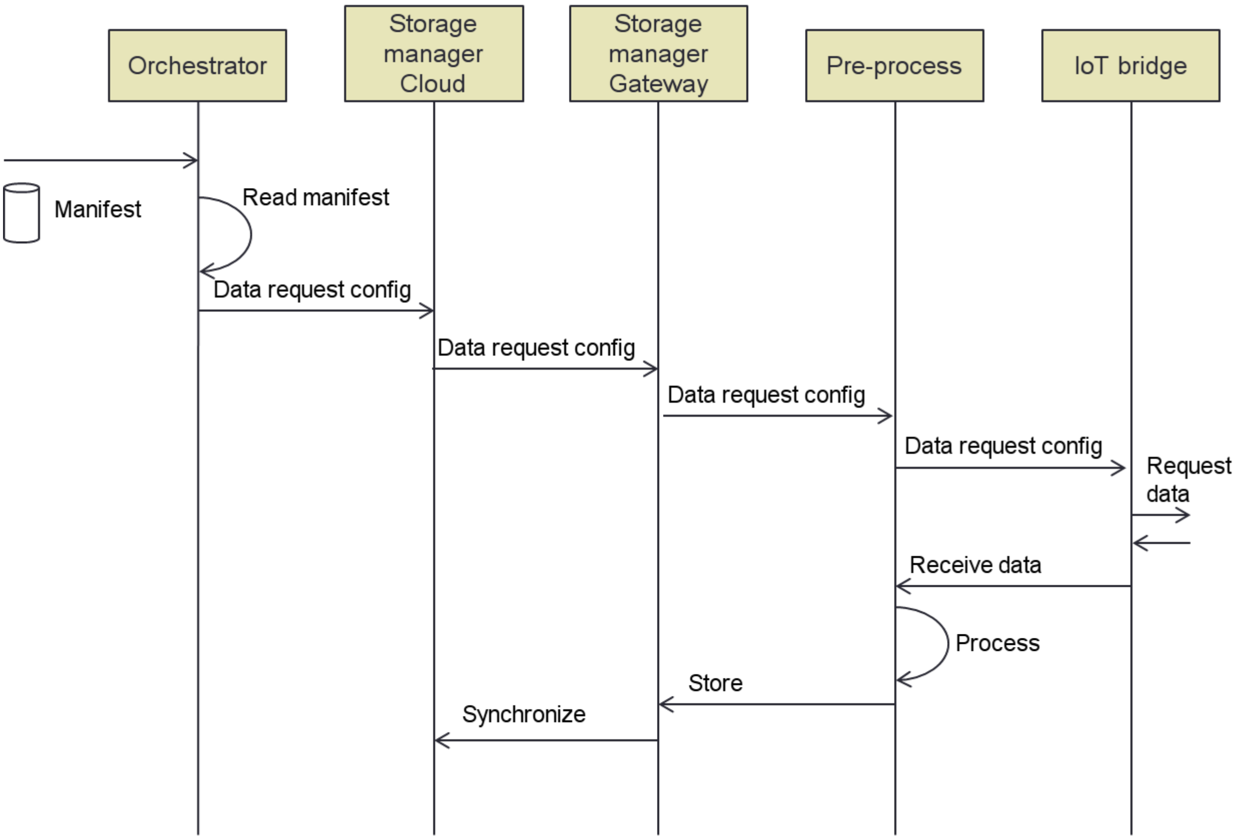


Figure 28: Pre-process and IoT bridge configuration diagram

By default, all the data collected by the sensors will be stored in the Cloud database. However, there is a mechanism to override this behaviour, and to configure more precisely the data streams, as shown in Figure 28. First, the orchestrator reads the data stream configuration expressed in the manifest. This data stream configuration is pushed on the Cloud Storage manager, which forwards it to the Gateway storage manager, the pre-process component and finally to the IoT bridge. The pre-process and IoT bridge as parametered using the configuration. The connection is then opened for the data stream in the reverse order: the IoT bridge receives the data, the Pre-process cleans and consolidates these raw data, the gateway Storage manager stores them. When the Cloud is available, data are synchronised between the Cloud Storage Manager and the gateway Storage Manager.

# 4 REQUIREMENTS

This section shows the list of engineering requirements that the WAZIUP platform should comply with. Each section shows the requirements for the components in a specific functional domain, namely the Application Platform domain, the IoT Platform domain, the Stream and Data Analytic domain, the Security and Privacy domain and finally the Platform Management domain. The definition of each functional domain and each component is given in Section 3. The requirements are separated in functional and non-functional requirements (labelled "F" and "NF" respectively). A functional requirement is feature oriented, whereas a non-functional requirement is quality and performance oriented. The sentence for each requirement is designed as follow: "The WAZIUP platform shall provide the end-user with the ability to compile an application".

The requirement elicitation process consists in deriving requirements from the use cases. This process is led by the architect and the engineers, with the help of the end users and SMEs. The requirements produced should be consistent, non-redundant, complete, unambiguous, testable, clear, correct, understandable, feasible, independent, atomic, necessary and finally implementation-free[4]. The objective of the requirement list is to describe precisely, for each of the components of the architecture, the detail of the features necessary. The requirement list provides an exact description of what features are required. However it shall not mention how (with which technologies) the features should be developed.

## 4.1 Application platform

This section shows the requirements for the "Application Platform" functional domain components, namely the Orchestrator, the Execution Environment, the Rapid Application Development and the UI manager. They can be found in Table 67.

Table 67: Application platform requirements

| ID | F/NF | Description | Component |
|---|---|---|---|
| App-10 | F | The WAZIUP platform shall be able to compile applications from their source code. | Orchestrator |
| App-20 | F | The WAZIUP platform shall provide the software libraries necessary to compile the application. | Orchestrator |
| App-30 | F | The WAZIUP platform shall support applications written in the following languages:<br>• Java<br>• Python<br>• Scala<br>• C/C++<br>• Golang | Orchestrator |

[4] http://www.ibmpressbooks.com/articles/article.asp?p=1152528&seqNum=4

| | | | |
|---|---|---|---|
| App-40 | F | The WAZIUP platform shall be able to read the manifest file provided by each application:<br>• Computation needs<br>• Sensing needs<br>• Big data engine<br>• GUI | Orchestrator |
| App-50 | F | The WAZIUP platform shall be able to provision the services described in the manifest file. | Orchestrator |
| App-60 | F | The WAZIUP platform shall be able to start, stop and restart applications | Orchestrator |
| App-70 | F | The WAZIUP platform should be able to schedule application tasks based on the manifest | Orchestrator |
| App-80 | F | The WAZIUP platform should be able to restart applications upon crash | Orchestrator |
| App-90 | F | The WAZIUP platform shall be able to deploy an application in the Cloud | Orchestrator |
| App-100 | F | The WAZIUP platform shall be able to deploy an application in the local platform (PC or Gateway) | Orchestrator |
| App-120 | F | The WAZIUP platform shall provide a way to specify the location of sensors to be selected in the manifest file | Orchestrator |
| App-130 | F | The WAZIUP platform shall be able to deploy the application. | Execution environments |
| App-140 | F | The WAZIUP platform shall be able to run the application. | Execution environments |
| App-150 | F | The WAZIUP platform shall support applications with several service components. | Execution environments |
| App-160 | F | The WAZIUP platform shall support applications that themselves expose APIs (microservices) | Execution environments |
| App-170 | NF | The WAZIUP platform shall provide an execution environment that is portable | Execution environments |
| App-180 | NF | The WAZIUP platform shall provide an execution environment that is independent from the system/hardware | Execution environments |
| App-190 | F | The WAZIUP platform shall provide a rapid application development (RAD) tool allowing the developer to create an application quickly | RAD |

| | | | |
|---|---|---|---|
| App-200 | F | The WAZIUP platform shall provide a RAD able to program data streams processing and events graphically | RAD |
| App-210 | F | The WAZIUP platform shall provide a RAD able to produce source code | RAD |
| App-220 | F | The WAZIUP platform shall provide a RAD able to program with source data streams coming from sensors | RAD |
| App-230 | F | The WAZIUP platform shall provide a RAD able to program with source data streams coming from internet APIs | RAD |
| App-240 | F | The WAZIUP platform shall provide a RAD able to program user outputs such as SMS and HTML pages | RAD |
| App-250 | F | The WAZIUP platform shall be able to spawn the GUI described in the manifest/application. | UI manager |
| App-260 | F | The WAZIUP platform shall provide access to the GUI of the application. | UI manager |
| App-270 | F | The WAZIUP platform shall be able to deploy mobile applications GUI. | UI manager |
| App-280 | F | The WAZIUP platform shall provide GUI templates to application developers. | UI manager |
| App-290 | F | The WAZIUP platform shall provide the following GUI functionalities:<br>• display reading<br>• set alarms | UI manager |
| App-300 | F | The WAZIUP platform shall provide GUI widgets for:<br>• Scalar display (e.g. number display, gauge…)<br>• Boolean display (e.g. checkbox, red/green lamp)<br>• Time series display (e.g. curves)<br>• Text display<br>• Picture/video display | UI manager |
| App-310 | F | The WAZIUP platform shall provide a SMS service | UI manager |
| App-320 | F | The WAZIUP platform shall provide a voice call service | UI manager |
| App-330 | F | The WAZIUP platform shall be able to deploy desktop applications GUI. | UI manager |
| App-340 | F | The WAZIUP platform shall provide degraded modes when connection to the Cloud is lost | N/A |

| | | | |
|---|---|---|---|
| App-350 | NF | The WAZIUP platform shall be able to manage 100 users (connections) simultaneously. | N/A |
| App-360 | NF | The WAZIUP platform shall be able to guarantee an application responsiveness under 500 ms. | N/A |
| App-370 | NF | The WAZIUP platform should be able to offer different level of complexities (based on the developer experiences) | N/A |

## 4.2 IoT Platform

This section gives the requirements for each components belonging to the "IoT Platform" functional domain, namely the IoT Bridge, the Pre-Process, the Sensor Registry and the Sensor Discovery components. They can be found in Table 68.

**Table 68: IoT Platform requirements**

| ID | F/NF | Description | Component |
|---|---|---|---|
| IoT-010 | F | The WAZIUP platform shall support sensors that have local processing capacity. | IoT bridge |
| IoT-020 | F | The WAZIUP platform shall support sensors that combine several physical observations (e.g. position + acceleration) | IoT bridge |
| IoT-030 | F | The WAZIUP platform shall support configuring the sensors over the network. | IoT bridge |
| IoT-040 | F | The WAZIUP platform shall support configuring the sensors over the air. | IoT bridge |
| IoT-050 | F | The WAZIUP platform shall support simple event-based mechanism from the sensors | IoT bridge |
| IoT-060 | F | The WAZIUP platform shall support the definition of the physical dimension of the values read from sensors (Temperature, Humidity etc). | IoT bridge |
| IoT-070 | F | The WAZIUP platform shall support the definition of the physical unit of the values read from sensors (Degree Celsius, Percentage, etc.) | IoT bridge |
| IoT-080 | F | The WAZIUP platform shall support the definition of relations between sensors, parameters and units. | IoT bridge |
| IoT-090 | F | The WAZIUP platform shall support the definition of relations between physical parameters and units (e.g. Temperature -> Degree Celsius, Humidity -> Percentage) | IoT bridge |
| IoT-100 | F | The WAZIUP platform shall support sensors with a sleep mode | IoT bridge |

| | | | |
|---|---|---|---|
| IoT-120 | F | The WAZIUP platform shall support interference avoidance mechanisms | IoT bridge |
| IoT-130 | F | The WAZIUP platform shall provide/support a naming service such as DNS | IoT bridge |
| IoT-140 | F | The WAZIUP platform shall support mobility of physical devices (mobile phones, sensor relocation) | IoT bridge |
| IoT-150 | F | The WAZIUP platform shall support IP communication protocols | IoT bridge |
| IoT-160 | F | The WAZIUP platform shall support IPv4 addressing service | IoT bridge |
| IoT-170 | F | The WAZIUP platform shall support monitoring of radio quality and usage | IoT bridge |
| IoT-180 | F | The WAZIUP platform shall support control of radio quality and usage | IoT bridge |
| IoT-190 | F | The WAZIUP platform shall support degraded modes able to save energy at sensor and gateway levels | IoT bridge |
| IoT-200 | F | The WAZIUP platform shall provide a uniform interface to access sensor data | IoT bridge |
| IoT-210 | F | The WAZIUP platform shall support constrained network stacks | IoT bridge |
| IoT-220 | F | The WAZIUP platform shall be able to put in energy saving mode a sensor that is not used by any application | IoT bridge |
| IoT-230 | NF | The WAZIUP platform shall be capable of providing reliable data transmission between the sensors, the gateway and the cloud platform. | IoT bridge |
| IoT-240 | NF | The WAZIUP platform shall support energy-efficient sensors able to run on battery for several months | IoT bridge |
| IoT-250 | NF | The WAZIUP platform shall support sensors with low amount of memory (or not at all) | IoT bridge |
| IoT-260 | NF | The WAZIUP platform shall support sensors working in the temperature range -40C to +50C | IoT bridge |
| IoT-270 | NF | The WAZIUP platform shall support lossy networks | IoT bridge |
| IoT-280 | NF | The WAZIUP platform shall support low bandwidth networks | IoT bridge |
| IoT-290 | NF | The WAZIUP platform shall support intermittent connection with the sensors | IoT bridge |

| | | | |
|---|---|---|---|
| IoT-300 | F | The WAZIUP platform shall be able to apply transformations to the data received from the sensors. | Pre-Process |
| IoT-310 | F | The WAZIUP platform shall be able to convert the data from the sensors to the internal data format. | Pre-Process |
| IoT-320 | F | The WAZIUP platform shall support the representation of physical location of objects | Sensor registry |
| IoT-330 | F | The WAZIUP platform shall be able to register new sensors. | Sensor registry |
| IoT-340 | F | The WAZIUP platform shall be able to unregister sensors. | Sensor registry |
| IoT-350 | F | The WAZIUP platform shall provide a format for sensor description (i.e. meta-data) | Sensor registry |
| IoT-360 | F | The WAZIUP platform shall support the representation and retrieval of the information quality from the sensors | Sensor registry |
| IoT-370 | F | The WAZIUP platform shall allow to set the measurement precision | Sensor registry |
| IoT-380 | F | The WAZIUP platform shall allow to set the measurement sampling rate | Sensor registry |
| IoT-390 | F | The WAZIUP platform shall enforce that each sensor have a unique ID | Sensor registry |
| IoT-400 | F | The WAZIUP platform shall maintain an updated list of relations between sensors and applications | Sensor registry |
| IoT-410 | F | The WAZIUP platform shall allow to query the list of sensors associated to a specific application | Sensor registry |
| IoT-420 | F | The WAZIUP platform shall allow to query the list of applications using a specific sensor | Sensor registry |
| IoT-430 | F | The WAZIUP platform shall allow a high-level language to describe queries on sensors meta-data | Sensor discovery |
| IoT-440 | F | The WAZIUP platform shall be able to return a list of sensors matching the query | Sensor discovery |
| IoT-450 | F | The WAZIUP platform shall be able to update the list of matching sensors based on the current status of sensors | Sensor discovery |
| IoT-460 | F | The WAZIUP platform shall support queries on:<br>• the location of sensors<br>• the owner<br>• the type of measurements | Sensor discovery |

## 4.3 Stream and data analytic

This section gives the requirements for the various components belonging to the "Stream and Data Analytic" functional domain, namely the Data Broker, the Storage Manager and the Data Analytic components. They can be found in Table 69.

**Table 69: Stream and Data Analytic requirements**

| ID | F/NF | Description | Component |
|---|---|---|---|
| Data-10 | F | The WAZIUP platform shall be able to collect data from third party data sources, such as internet web services. | Data broker |
| Data-20 | F | The WAZIUP platform shall support publish-subscribe data distribution model | Data broker |
| Data-30 | F | The WAZIUP platform shall allow applications to subscribe on events | Data broker |
| Data-40 | F | The WAZIUP platform shall allow applications to publish events | Data broker |
| Data-50 | | The WAZIUP platform shall support quality of information support in the event processing | Data broker |
| Data-60 | F | The WAZIUP platform shall support applications that produce data consumed by other applications | Data broker |
| Data-70 | F | The WAZIUP platform shall provide an event processing engine. | Data broker |
| Data-80 | F | The WAZIUP platform shall allow applications to retrieve information from external databases and APIs | Data broker |
| Data-90 | F | The WAZIUP platform shall support processing of events on a distributed event engine | Data broker |
| Sens-100 | F | The WAZIUP platform shall be able to read the status of sensors | Data broker |
| Data-110 | F | The WAZIUP platform shall provide a unified interface to the data stores, local and global. | Storage manager |
| Data-120 | F | The WAZIUP platform shall be able to store the information related to users: developers, sensor owners and app user. | Storage manager |
| Data-130 | F | The WAZIUP platform shall be able to store the data generated by an application | Storage manager |
| Data-140 | F | The WAZIUP platform shall be able to store the information related to sensors (meta data) | Storage manager |

| | | | |
|---|---|---|---|
| Data-150 | F | The WAZIUP platform shall be able to store the data collected by sensors | Storage manager |
| Data-160 | F | The WAZIUP platform shall be able to manage data replication between local deployment and Cloud | Storage manager |
| Data-170 | F | The WAZIUP platform shall be able to synchronize the storage of local data and cloud data | Storage manager |
| Data-180 | F | The WAZIUP platform shall be able to ensure the persistence of data (in local) even if we are out of connection | Storage manager |
| Data-190 | F | The WAZIUP platform shall be able to delete data from data storages | Storage manager |
| Data-200 | F | The WAZIUP platform shall be able to manage the lifecycle of data, from creation to deletion | Storage manager |
| Data-210 | F | The WAZIUP platform shall be able to manage expiry date for data validity | Storage manager |
| Data-220 | F | The WAZIUP platform shall be able to support scalability by allowing data replication and distribution | Storage manager |
| Data-230 | F | the WAZIUP platform shall be able to ensure fault tolerance | Storage manager |
| Data-240 | F | The WAZIUP platform shall enable handling of different structural data | Storage manager |
| Data-250 | F | The WAZIUP platform shall be able to handle applications data on the basis of unique data format | Storage manager |
| Data-260 | F | The WAZIUP platform shall be able to provide a database service to applications in order to store historical data. | Storage manager |
| Data-270 | F | The WAZIUP platform shall provide mechanisms for application deployed in the local platform to access data stored locally | Storage manager |
| Data-280 | F | The WAZIUP platform shall be able to store sensor data in an historical database | Storage manager |
| Data-290 | F | The WAZIUP platform shall be able to remove sensor data from the historical database. | Storage manager |
| Data-300 | F | The WAZIUP platform shall be able to aggregate/consolidate sensor data into bigger structures. | Storage manager |
| Data-310 | F | The WAZIUP platform shall be able to provide pattern analysis services. | Data analytic |

| | | | |
|---|---|---|---|
| Data-320 | F | The WAZIUP platform shall be provide trend prediction analysis services based on historical data. | Data analytic |
| Data-330 | F | The WAZIUP platform shall provide machine learning service:<br>• Data pre-processing (e.g. feature extraction, dimensionality reduction)<br>• Supervised learning<br>• Unsupervised learning | Data analytic |
| Data-340 | NF | The WAZIUP platform shall be able to manage 1000 data streams simultaneously. | N/A |
| Data-350 | NF | The WAZIUP platform shall guarantee the delivery of messages to subscribers under 100 ms | N/A |

## 4.4 Security and privacy

This section gives the requirements for each components belonging to the "Security and privacy" functional domain, namely the Identity Manager, the Authorization Manager and the Privacy Manager. They can be found in Table 70.

**Table 70: Security and Privacy requirements**

| ID | F/NF | Description | Component |
|---|---|---|---|
| Sec-10 | F | The WAZIUP platform shall provide a secure access to remote and local resources. | Identity manager |
| Sec-20 | F | The WAZIUP platform shall provide mechanisms to make the local system available. | Identity manager |
| Sec-30 | F | The WAZIUP platform shall provide authentications mechanisms to users. | Identity manager |
| Sec-40 | F | The WAZIUP platform shall ensure authentication on the gateway level. | Identity manager |
| Sec-50 | F | The WAZIUP platform shall ensure a secure communication between the devices and the gateway and between the gateway and the cloud. | Identity manager |
| Sec-60 | F | The WAZIUP platform shall provide mechanisms to secure the access to user devices. | Identity manager |
| Sec-70 | F | The WAZIUP platform shall be able to allow a user to register with the platform. | Identity manager |
| Sec-80 | F | The WAZIUP platform shall define the following roles for the | Identity manager |

| | | | |
|---|---|---|---|
| | | users:<br>• Developer<br>• Sensor owner<br>• Application users<br>• Platform administrator | |
| Sec-90 | F | The WAZIUP platform shall support applications with login and passwords of users. | Identity manager |
| Sec-100 | F | The WAZIUP platform shall allow a user to unregister from the platform. | Identity manager |
| Sec-110 | F | The WAZIUP platform shall allow the administrator to take action about the users:<br>• Add user<br>• Remove user | Identity manager |
| Sec-120 | F | The WAZIUP platform shall allow the administrator to inherit the access rights of any users. | Identity manager |
| Sec-130 | F | The WAZIUP platform shall not disclose publicly the access control policies set up by data owners. | Authorization manager |
| Sec-140 | F | The WAZIUP platform shall provide mechanisms to manage the access control policies to the resources and services in the cloud and locally. | Authorization manager |
| Sec-150 | F | The WAZIUP platform shall ensure a confidential communication with the services. | Privacy manager |
| Sec-160 | F | The WAZIUP platform shall ensure communication integrity while accessing the services. | Privacy manager |
| Sec-170 | F | The WAZIUP platform shall be able to provide a data anonymization service. | Privacy manager |

## 4.5 Platform Management

The Platform Management functional domain covers the management of the WAZIUP platform itself, and notably its deployment and status. The corresponding requirements are given in the Table 71.

**Table 71: Platform Management requirements**

| ID | F/NF | Description | Component |
|---|---|---|---|
| PF-10 | F | The WAZIUP platform shall be able to manage its own status and the status of its components. Status are: | Status manager |

| | | | |
|---|---|---|---|
| | | • started<br>• stopped | |
| PF-20 | F | The WAZIUP platform shall be able to maintain a log of the status of its components | Status manager |
| PF-30 | F | The WAZIUP platform shall support auto-scaling of its resources if the workload become too important or the platform too slow. | Status manager |
| PF-40 | F | The WAZIUP platform shall give a high level view of the platform status to administrator:<br>• Sensor status<br>• Applications status<br>• User status | Status manager |
| PF-50 | F | The WAZIUP platform shall be able to run within an IoT simulator for development and debugging purpose. | N/A |
| PF-60 | NF | The WAZIUP platform shall be deployable on a single Linux PC, for development and testing purposes. | N/A |
| PF-70 | NF | The WAZIUP platform shall be deployable on a IaaS or PaaS Cloud platform. | N/A |
| PF-80 | NF | The WAZIUP platform shall use lightweight containers | N/A |
| PF-90 | NF | The WAZIUP platform shall be deployed in a local deployment, such as PC or gateway | N/A |
| PF-100 | NF | The WAZIUP platform shall provide an affordable access and price plan | N/A |
| PF-110 | NF | The WAZIUP platform shall manage intermittent internet connection | N/A |
| PF-120 | NF | The WAZIUP platform shall be able to run with the minimum configuration:<br>- CPU 2 GHz<br>- RAM 4 Go<br>- HD 40 Go | N/A |
| PF-130 | NF | The WAZIUP platform shall be able to manage 100 user connections simultaneously. | N/A |

# 5 INTERFACES

This section describes the external and internal interfaces provided by the WAZIUP platform. In the first section, the WAZIUP platform will be seen as a single "black box", with external connections. In the section, all internal components interfaces will be described.

## 5.1 External interfaces

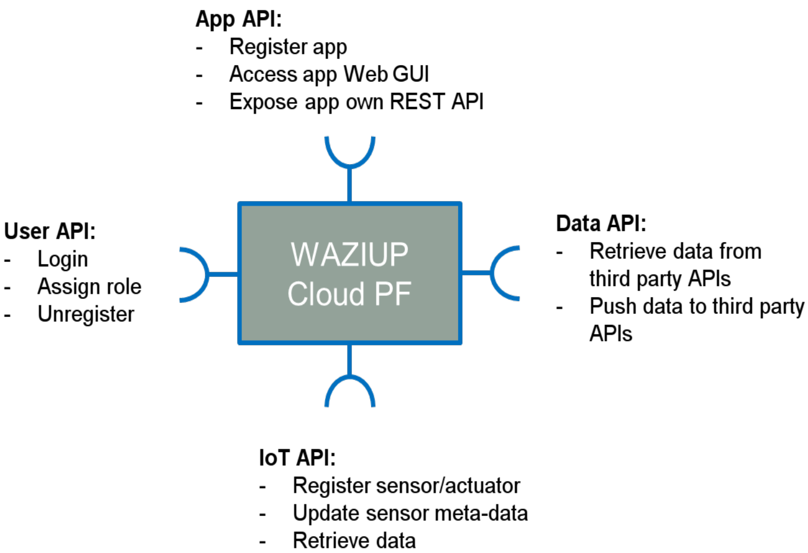


Figure 29: WAZIUP Cloud platform external interfaces

The Figure 29 shows the four external interfaces of WAZIUP Cloud platform: the Application API, User API, Data API and finally the IoT API. The first API is dedicated to inserting and accessing applications in the WAZIUP platform. The User API allows end users to identify and access the resources. The Data API is devoted to the communication with third-party data providers, like weather informations. Finally the IoT API manages the communication with the physical devices. This includes the registration of sensors, the meta-data settings (such as sensor position, type and owner) and the data interface for actual measurements and actuations. The Table 72 shows the detail of the external APIs.

Table 72: WAZIUP external APIs

| API name | API provided |
|---|---|
| **App API** | • App registration interface<br>• App GUI access: web and mobile<br>• Expose App own REST API<br>• App management interface to start, stop and restart an app<br>• App templates and widgets |
| **User API** | • Registration interface of a new user<br>• Login interface of a registered user<br>• Roles Assignment: Developer, Sensor owner, Application users, Platform administrator |

| | |
|---|---|
| | • Un-registration interface of an existing user<br>• User information interface |
| **Data API** | • Retrieve data from the platform<br>• Retrieve data from third party<br>• Push data to the platform (from third party) |
| **IoT API** | • Register sensor/actuator<br>• Update sensor meta-data (configure the sensor)<br>• Retrieve sensors data<br>• Unregister a sensor/actuator<br>• Send action to an actuator<br>• Provide subscription/notification mechanisms to sensors data |

## 5.2 Internal interfaces

This section details the interfaces for each component. The list of components and their function is visible in Section 3. In the following, we show the interfaces of each components for the 4 functional domains of WAZIUP: "Application platform", "IoT platform", "Stream and data analytic" and finally "Security and Privacy".

### 5.2.1 Application platform components

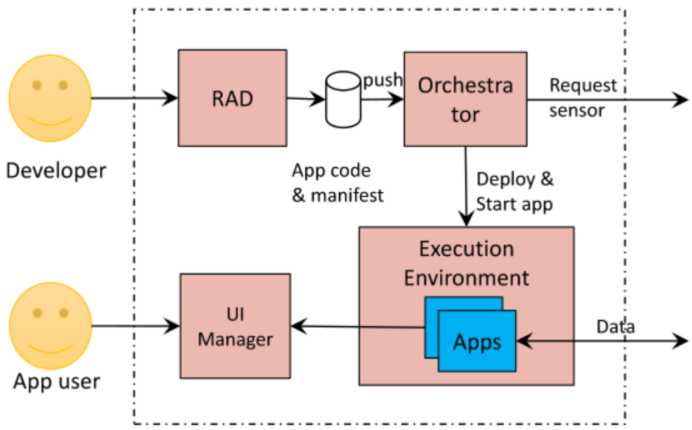


Figure 30: Application platform interfaces

The first component is the Orchestrator. Its role is to read the manifest, trigger the compilation and deployment of the applications, instantiate the services needed by the applications and finally request the sensor and data sources connections. The execution environment hosts the various containerized applications. The Rapid Application Development system provides a graphical user interface and produces executable code. It does not expose an API. The UI manager exposes APIs allowing the user application to create user interfaces on the fly. The UI manager exposes several APIs, allowing creating Web dashboards, SMS/USSD and Voice calls. All the APIs are detailed in Table 73.

Table 73: Application Platform APIs

| Component name | API provided | API requested from other components |
|---|---|---|
| **Orchestrator** | • Create App (from code + manifest)<br>• Get the list of applications<br>• Delete an app<br>• List app details<br>• Update app details<br>• Deploy app<br>• Start app<br>• Stop app<br>• Scale app | Execution Environment: all the API |
| **Execution Environment** | • Create container<br>• Delete container<br>• List containers<br>• Start container<br>• Stop container<br>• Scale container | None |
| **RAD** | none | None |
| **UI manager** | Web dashboards:<br>• New dashboard<br>• Create widget<br>• Set datasource<br>• Attach callback<br>SMS/USSD:<br>• Send SMS<br>• Set menu items<br>• Attach callback<br>Voice call:<br>• Send voice call<br>• Set menu item<br>• Attach callback | None |

### 5.2.2 IoT platform components

The interaction between the IoT platform components is shown in Figure 31.

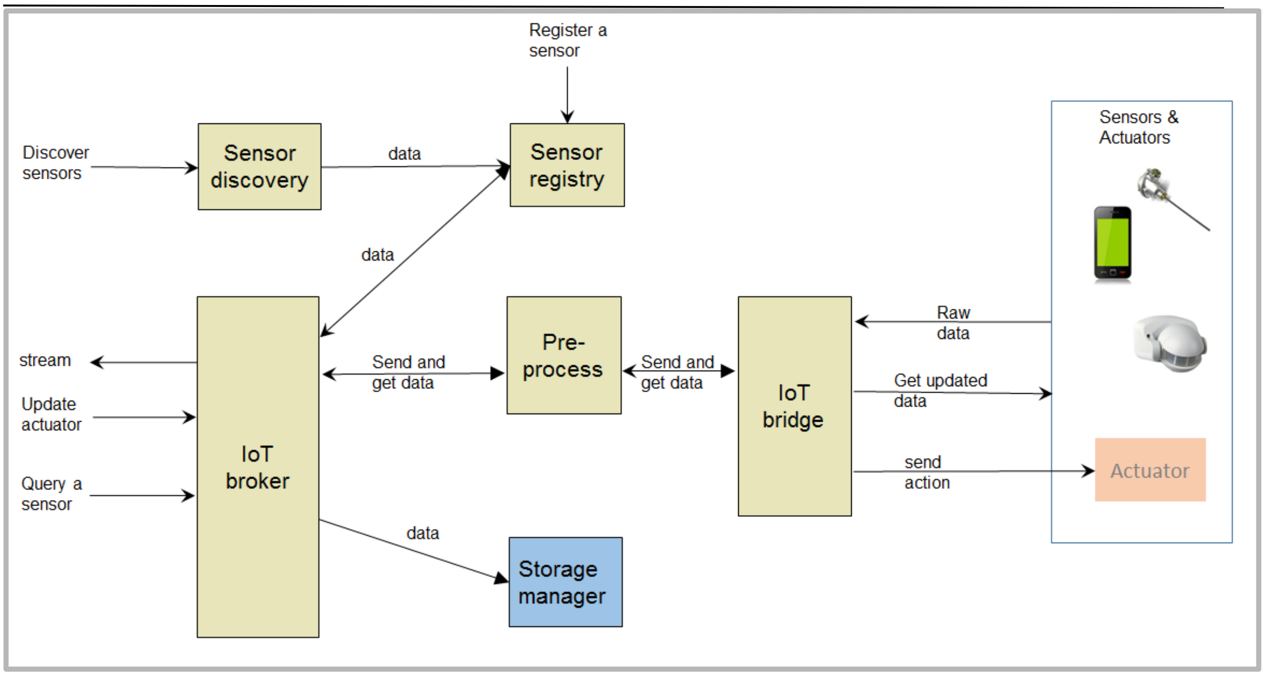


**Figure 31 IoT platform interfaces**

First al all, the role of the IoT bridge is to ensure the connectivity between the sensors and the Platform. It is able to receive the sensor row data using the sensor protocol. It can also receive commands from components, which are requests to send to the sensor/actuator and request to give the late value updated by the sensor. It then sends the raw data to the Pre-Process module to be processed. The Pre-Process component processes the received data (transform its format, apply some operator, etc). The sensor registry role is to store the virtual representation of the real sensor. The sensor discovery role is to find the list of available sensors that matches a query. All the APIs are detailed in Table 74. The IoT broker API is described in the Stream and Data Analytic section.

**Table 74: IoT platform APIs**

| Component name | API provided | API requested from other components |
|---|---|---|
| **IoT bridge** | • Receive data from sensor<br>• Receive command from a component | • Sensor API: get updated data<br>• Actuator API: execute action<br>• Pre-Process API: receive data from components |
| **Pre-Process** | • Receive data from the components | • IoT bridge API: receive command from a component<br>• IoT broker API: receive sensor updates |
| **Sensor registry** | • Register virtual entity representing the sensor<br>• Unregister virtual entity representing the sensor<br>• Query the virtual entity representing the sensor | None |

| | | |
|---|---|---|
| | • Update the virtual entity representing the sensor<br>• Query the data | |
| **Sensor discovery** | • Specify the query | • Sensor registry API: query the data |

### 5.2.3 Stream and data analytic

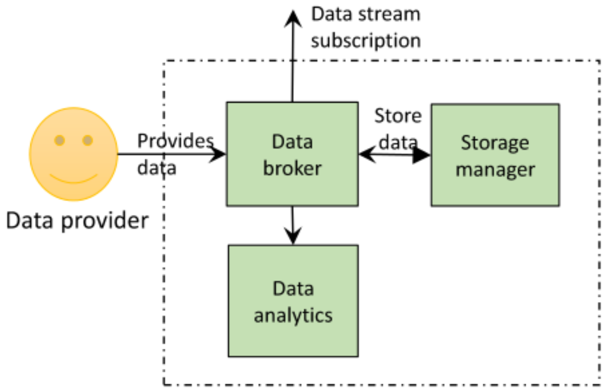


Figure 32: Stream and data analytics interfaces

As can be seen in Figure 32, the data broker is the central component of the platform. It manages information about the context. It is able to receive and provides context information. It also provides a messaging system based on subscription. The main role of the Storage Manager is to provide interface to store and retrieve data. It must also take into account the heterogeneity aspect and performance issues concerning real-time queries with sensor data. The role of the Data Analytic component is to provide advanced data processing operations to the user applications. All the APIs are detailed in Table 75.

Table 75: Stream and data analytic APIs

| Component name | API provided | API requested from other components |
|---|---|---|
| **Data broker** | • Receive sensor updates<br>• Query a sensor data<br>• Send action to an actuator<br>• Subscribe to a sensor data | • Pre-Process API: receive data from the components<br>• A Subscribed Component API: get notifications (to receive the streamed data)<br>• Storage manager: store data<br>• Sensor registry: get the virtual entity representing the sensor<br>• Sensor registry: update the virtual entity representing the sensor |
| **Storage** | • Store data related to users | None |

| | | |
|---|---|---|
| **Manager** | (personal and login data)<br>• Store data related to applications (app data and app generated data)<br>• Store data related to sensors (sensor meta-data and sensed data)<br>• Trigger a backup<br>• Update database schema<br>• Retrieve data related to entities<br>• Delete data related to entities | |
| **Data analytics** | • Create model for dynamics estimations<br>• Trigger event (alarm)<br>• Recognize simple event (e.g., out of range)<br>• Recognize complex event (e.g. soil conditions change)<br>• Predict situation using model for decision and actuation | None |

## 5.2.4 Security and privacy

As can be seen in Figure 33, the Service and Privacy domain has the following components:

- Identity manager
- Authorization manager
- Privacy manager

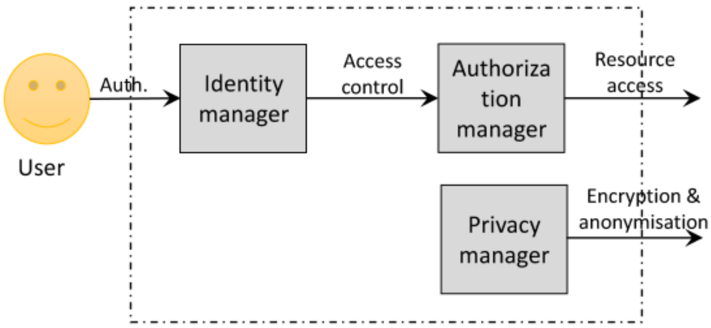


Figure 33: Security and privacy interfaces

The Identity Manager provides a centralized service to identify users, based on login/password or more advanced protocols. The Authorization Manager provides services to control the access to the resources. The Privacy Manager provides a data anonymization service as well as data encryption service. All the APIs are detailed in Table 76.

Table 76: Security and Privacy APIs

| Component name | API provided | API requested from other components |
|---|---|---|
| **Identity Manager** | • Register new user<br>• Unregister existing user<br>• Request authentication | None |
| **Authorization Manager** | • Add access control rule<br>• Update access control rule<br>• Get access control rule<br>• Delete access control rule<br>• Request authorization | None |
| **Privacy Manager** | • Anonymize data<br>• Encrypt data | None |

# 6 IOT PLATFORM, INFRASTRUCTURE AND DEPLOYMENT

This section describes the IoT platform and infrastructure, from the hardware point of view. It first proposes an overview of the techniques used, and then shows the detailed architecture.

## 6.1 Overview

During the last decade, low-power but short-range radio such as IEEE 802.15.4 radio coupled with 6LowPan/RPL capabilities for instance have been considered by the WSN community with multi-hop routing to overcome the limited transmission range, as shown in Figure 34. While such short-range communications can eventually be realized in the context of developed countries smart cities infrastructures where high node density with powering facility can be achieved, it can hardly be generalized for the large majority of surveillance applications that need to be deployed in isolated or rural environments.

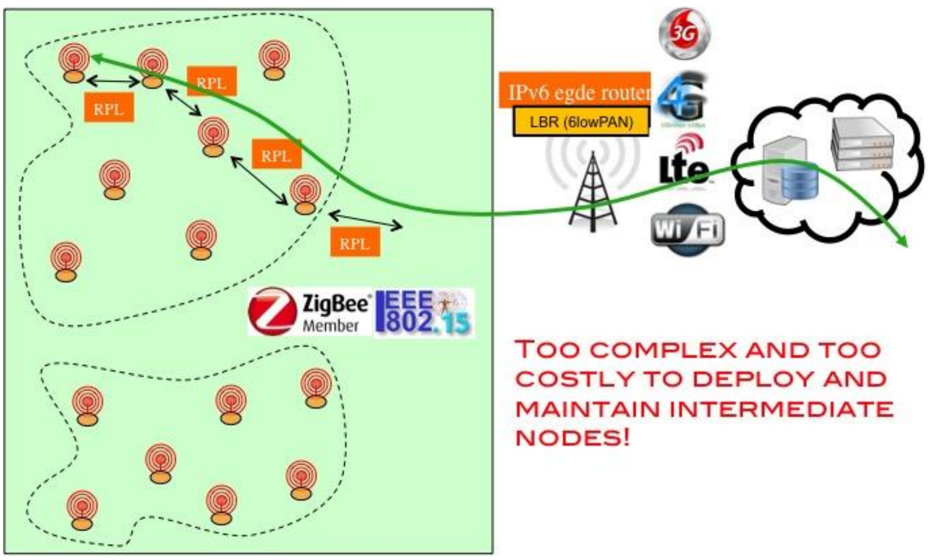


Figure 34: Multi-hop transmission

WAZIUP mainly promotes the usage of low-power, long-range WAN (LPWAN) and operator-free radio technologies to connect deployed sensors at the lowest financial cost. LPWAN provides promising technologies for low-throughput, long-range Machine-to-Machine/IoT connectivity as illustrated in Figure 35.

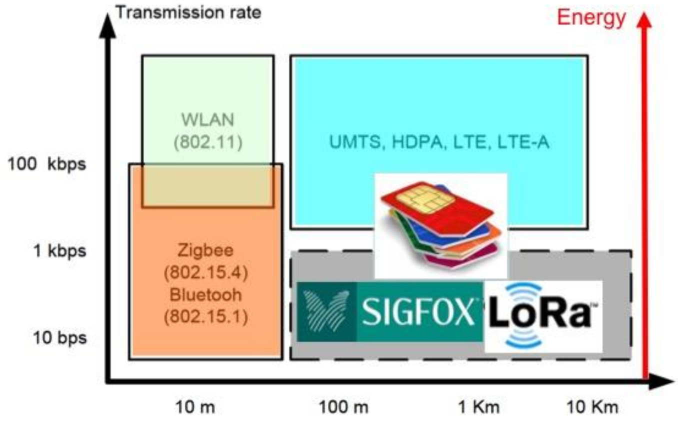


Figure 35: Transmission rate vs distance

Recent so-called Low-Power Wide Area Networks (LPWAN) such as those based on Sigfox(™) or Semtech's LoRa(™) technology definitely provide a better connectivity answer for IoT as several kilometers can be achieved without relay nodes to reach a central gateway or base station while keeping the transmission energy consumption comparable to short-range radio technologies (see Figure 36).

Tables from Semtech

| Technology | 2G | 3G | LAN | ZigBee | Lo Power WAN |
|---|---|---|---|---|---|
| Range (I=Indoor, O=Outdoor) | N/A | N/A | O: 300m I: 30m | O: 90m I: 30m | Same as 2G/3G |
| Tx current consumption | 200-500mA | 500-1000mA | 100-300mA | 18mA | 18mA |
| Standby current | 2.3mA | 3.5mA | NC | 0.003mA | 0.001mA |
| Energy harvesting (solar, other) | No | No | No | Possible | Possible |
| Battery 2000mAh (LR6 battery) | 4-8 hours(com) 36 days(idle) | 2-4 hours(com) X hours(idle) | 50 hours(com) X hours(idle) | 60hours (com) | 120 hours(com) 10 year(idle) |
| Module Revenue Annually | 12 $ | 20 $ | 4 $ | $3 | 3 $ |

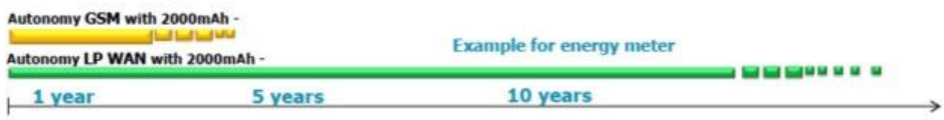


Figure 36: Radio technologies comparison

When adding the financial cost constraint and the network availability, LoRa technology, which can be privately deployed in a given area without any service subscription, has a clear advantage over Sigfox which coverage is entirely operator-managed.

Using LPWAN technologies, the network topology is mainly gateway-centric where an LPWAN gateway will be used to receive and collect sensed data in a single-hop model, thus greatly simplifying deployment. The LPWAN gateway must use the same radio technology than the one deployed in the sensors. Figure 37 shows a typical extreme long-range 1-hop connectivity scenario to a long-range gateway which is the single interface to Internet servers. Most of long-range technologies can achieve 20km or higher range in LOS condition and about 2km in NLOS. The LPWAN gateway will then be connected to the WAZIUP platform using traditional (Internet) networking protocols. In most cases, the gateway has Internet connectivity based on standard connectivity solutions, such as cellular (GSM/3G/4G), WiFi and Ethernet. As illustrated in the figure the IoT architecture consists in the (1) sensor nodes and (2) the gateway platform. Both platforms will interact with the long-range radio technology.

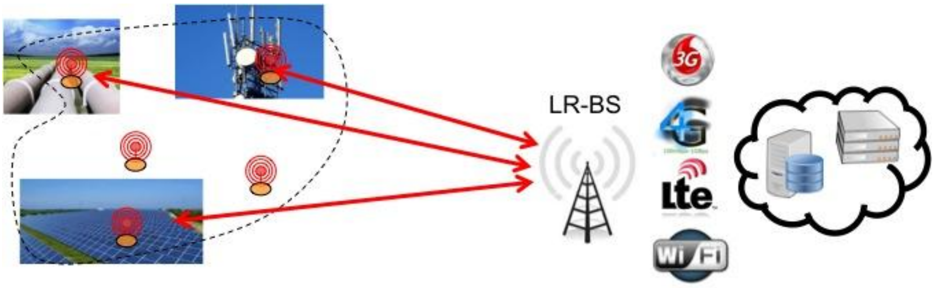


Figure 37: Long range single hop scenario

## 6.2 IoT platform architecture design

To address the low-cost issue, both Sensor and Gateway platforms will use off-the-shelves components. This section will describe the architecture of the sensors nodes, the gateway platforms and the network of WAZIUP.

### 6.2.1 Sensor nodes

WAZIUP has several Use Cases that focus on completely different issues, therefore imposing the need for WAZIUP to be very flexible in terms of the Hardware required to tackles each Use Case. So, taking this into account, WAZIUP needs to set up a way so that each piece of hardware deployed can be very specific to the particular Use Case, so that WAZIUP can reduce the costs to a minimum.

To tackle this, there is the need to understand what comprises the Hardware for sensing the environment. So, this devices, here called “IoT Devices”, need to be very flexible in terms of processing needs, in terms of sensing capabilities and also in terms of communication technologies used. To enable that the IoT Devices can be assembled using different pieces of hardware, a reference architecture was developed which is constituted by several different modules that need to be deployed on the IoT Devices so that it can work properly.

The IoT Device in WAZIUP is the device that will be monitoring the environment, collecting and processing the data necessary for the Use Cases and then communicates them to the WAZIUP Platform. So, it will have communication, processing and sensing capabilities, that means an IoT Device can measure one or more parameter from the environment, processing those parameters by generating meaningful data and then communicate the data by (possible) different means (radio, cable link, local storage, etc). The reference architecture for the IoT Devices is depicted in Figure 38. The reference architecture contains 3 different layers, and each one with a set of modules that are needed to represent the complete functionality of the IoT Devices.

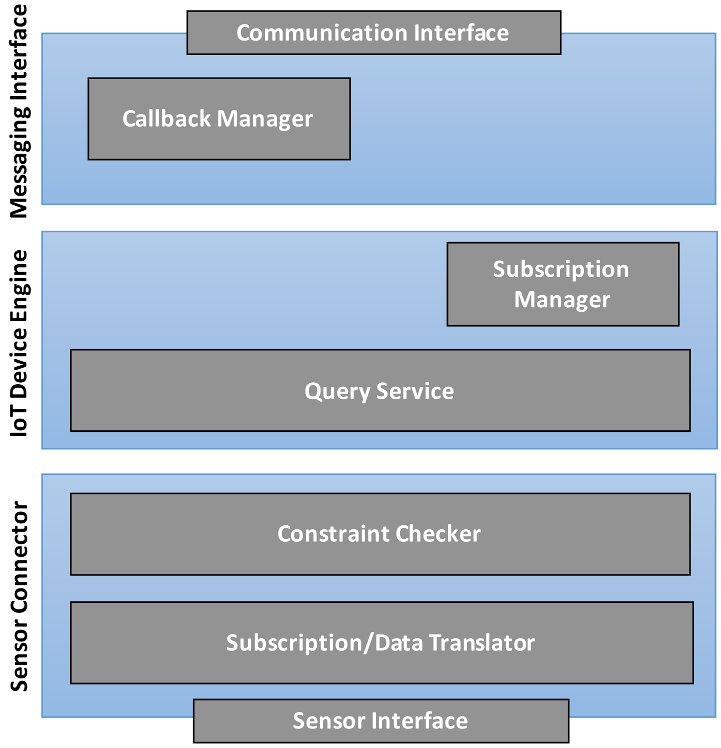


Figure 38: Architecture schema

In the following paragraphs, the reference architecture will be described, including a definition about each module, how each can be used on an IoT Device and the different functionalities each module provides.

The Sensor Connector layer aggregates the set of modules needed by the hardware related to sensing the environment, such as the sensor itself and the Sensor Interface (in case of need). The modules present in this layer are responsible for the interface with the real world in terms or measuring specific parameters from the environment, but also for interpret the data and to check for its validity. The modules present in this layer are:

- **Sensor Interface:** The Sensor interface is the module that actually connects to the sensor itself and retrieves the raw data from it.
- **Subscription/Data Translator:** This module is responsible for requesting the data from the Sensor Interface, and also receives the data from it. It then translates the raw data received to meaningful data.
- **Constraint Checker:** The Constraint Checker module validates if the request for data to the sensor is valid according to the attached sensor. It also validates if the data received from the sensor is what it is expected.

The IoT Device Engine layer is responsible for the management and processing of data on the IoT Device. Both the data obtained by the sensors and the requests received from the WAZIUP Platform are here analysed. This layer can have a variable processing power, depending on each of the Use Cases needs and consequently on the hardware chosen for the IoT Device. This layer presents the following modules:

- **Query Service:** This module refers to ability of querying the sensor for new data. This service manages everything related to the query service, including the number of requests, the time interval between requests, etc.

- **Subscription Manager:** This module is responsible for the subscription of the services provided by the IoT Device. The services provided by the IoT Device are dependent on the Use Case it is being used for, and also on the hardware chosen.

The Messaging Interface layer represents the capability of the IoT Device to communicate to the other WAZIUP Devices. This layer contains the modules that are necessary not only to send the data to the other WAZIUP devices but to receive the requests for data from them. This layer, is of course dependent on the Use Case and more precisely on the communication constraint that each Use Case have. The modules present within this layer are the following:

- **Callback Manager:** This module is responsible for setting up the communication messages, so that, when the IoT Device gathered the requested data, can send it to the correct WAZIUP device (that requested that specific data).
- **Communication Interface:** This module, is as the name suggests, the responsible for the setting up everything related to the communications between the IoT device and the other WAZIUP devices. This communication interface is largely dependent on the hardware chosen to communicate the IoT Device with the other WAZIUP devices.

This architecture will use data of several kinds (eg. water measurements, GPS coordinates, etc), which will be organized into domains where each one contains information about different sensors, the following paragraphs will present the concepts which describes how the data is organized.

### 6.2.2 IoT Device Components

Each domain has its own specific characteristics with different types of data, which are going to be obtained by the IoT Devices through measurements. These IoT Devices are composed by a group of hardware components, such as a board, sensor, radio and interfaces (when it's needed). Each IoT Device can have multiple sensors (and sensor interfaces) and also multiple radios (and radio interfaces), as far as the board supports it. The domains in WAZIUP rely on data gathered from the environment by the IoT Devices, and to be able to do so, the IoT Devices need to be composed by specific hardware:

- **Sensor:** used to collect data of physical parameters from the environment (e.g. Temperature, Humidity, etc.).
- **Interface:** Required by some hardware to communicate (eg, converting sensor conductivity into an analog signal).
- **Board:** Main component of an IoT Device. It's used to receive the data gathered from the sensor(s), process it as defined and forward it using the radio(s) available.
- **Radio:** This component is used for communication purposes, so that the IoT Device can receive requests and send data from/to the WAZIUP Platform.

In terms of readings from the IoT Device, it uses the sensor (or each one, in case of having more than 1) to gather data. This process is made through measurements, which are composed by the following elements:

- **Parameter:** Physical input that is read from the environment (e.g., Temperature, Humidity, etc.).
- **Unit:** How the data obtained by the measurement is represented (e.g., °F, °C).

Figure 39 shows the overall picture regarding the hardware components of an IoT Device and how it relates to the WAZIUP domains and to the measurements gathered form the environment.

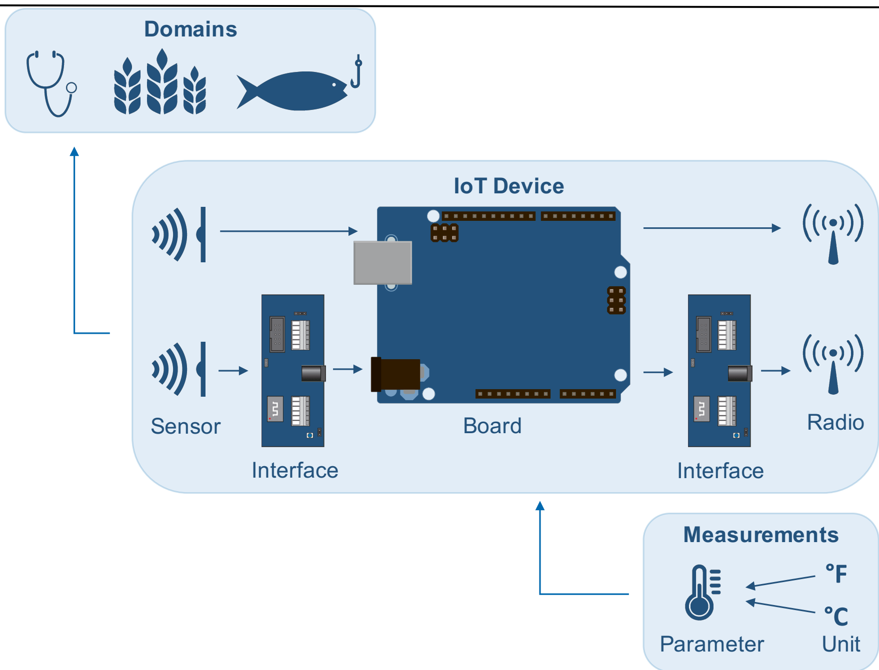


Figure 39: IoT Device Integration

The following example describes a use case from the "Fish Farming" domain, presenting an IoT device that can be used on this situation. The main purpose of the IoT Device here specified is to collect data from the water. Among several parameters, the water temperature (°C) is measured by using an IoT Device composed by the following hardware:

- **PT-1000 Temperature Sensor:** It's conductivity changes with the water temperature.
- **PT-1000 Temperature Module:** It translates the sensor conductivity into a temperature value
- **Arduino UNO Board:** Main board of the IoT Device
- **Lora RF Module:** Enables the communication from/to the IoT Device

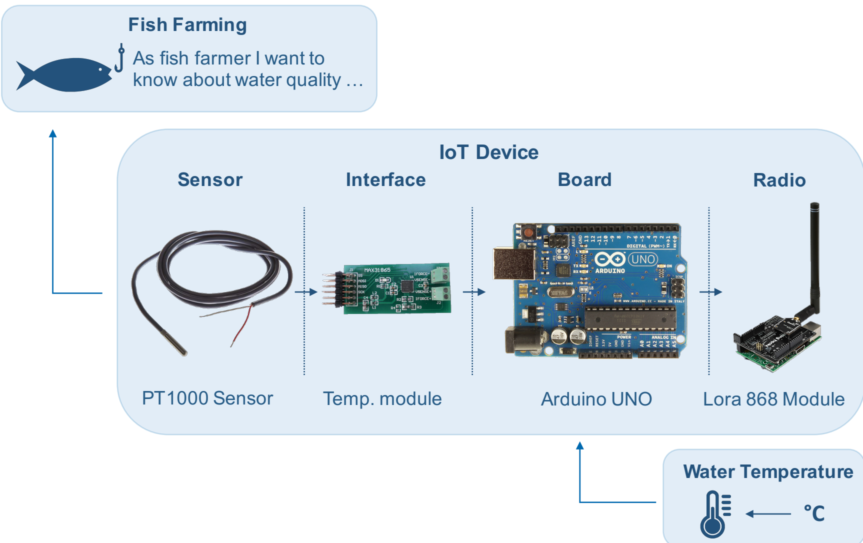


Figure 40: Fish Farming

The Arduino Pro Mini comes in a small form factor and is available in a 3.3v and 8MHz version for lower power consumption. It appears to be the development board of choice for providing a generic platform for IoT sensing as illustrated in Figure 41.

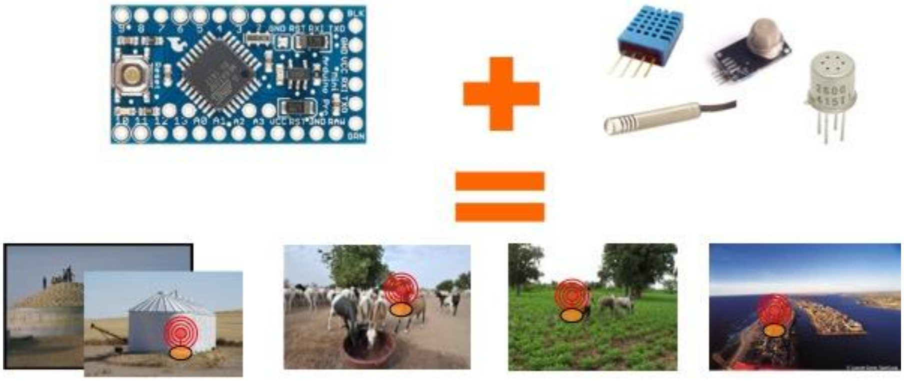

Figure 41: Arduino Mini in WAZIUP deployments

Adding specific physical sensor devices must be done on a particular use case or application basis. However, a generic template will be provided and used by the developer to quickly implement the physical and application specific part of the sensor nodes. Management of all the physical sensors, activity duty-cycling/low power mechanisms, security, long-range transmission and logical sensor management will be provided by WAZIUP's template as illustrated in Figure 42. Note that WAZIUP can also provide for some sensors ready-to-use templates for the physical sensor reading logic.

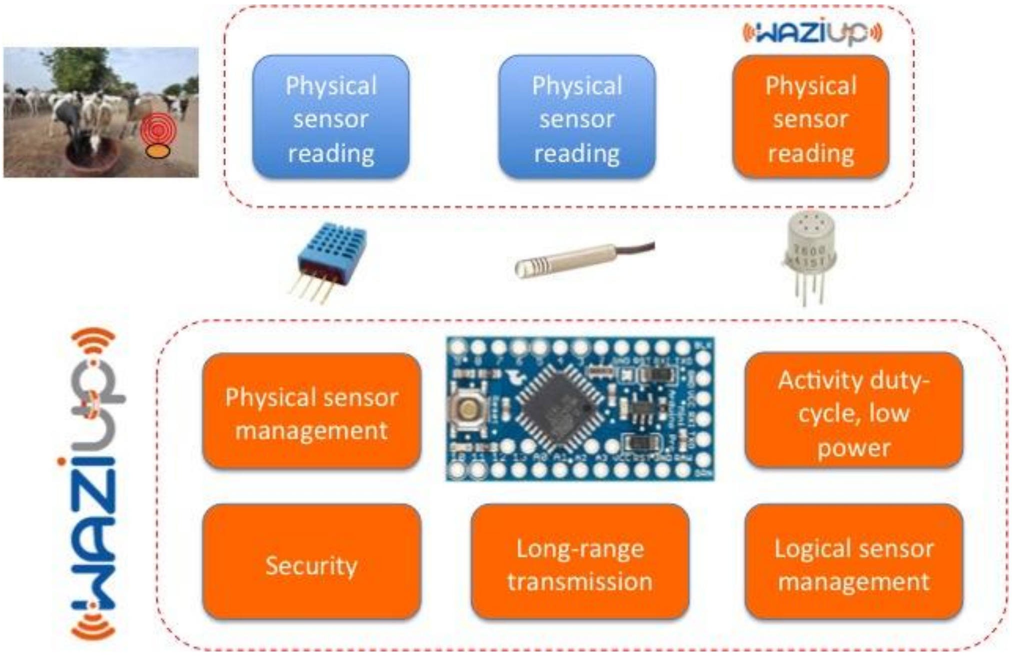


Figure 42: WAZIUP templates

Adding long-range transmission features will be provided by a low-level generic communication library such as illustrated by Figure 43: the communication library should be able to run for various mass-market microcontroller boards depending on the power and energy requirements/constraints of the final application.

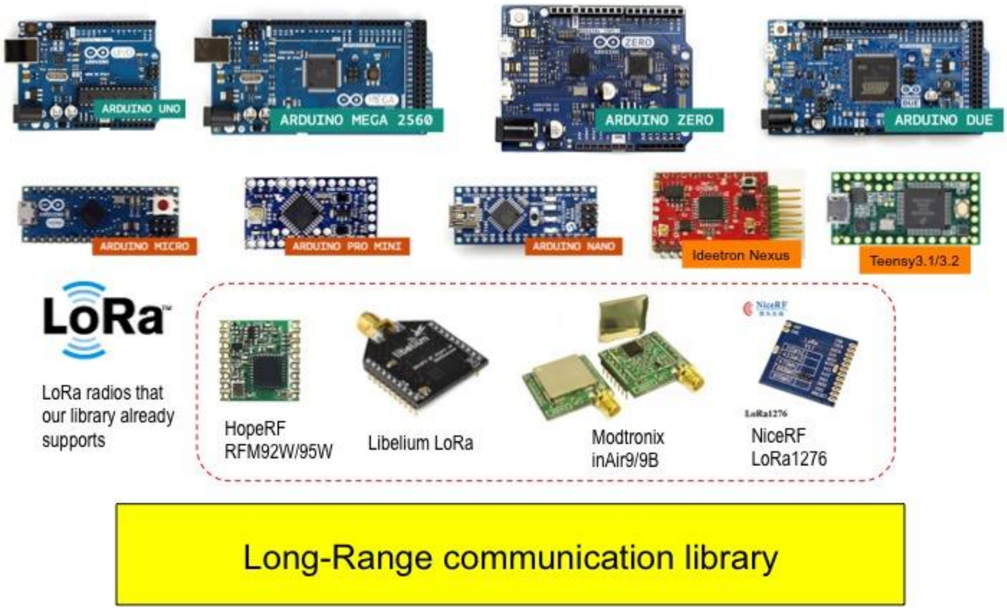


**Figure 43: Arduino LoRa library**

### 6.2.3 Gateway platform

With the gateway-centric mode of LPWAN technology, commercial LPWAN gateways are usually able to listen on several channels and transmission settings simultaneously. For instance, a LoRa gateway uses advanced concentrators chips capable of scanning up to 8 different channels: the SX1301 concentrator is typically used instead of the SX127x chip serie which is designed for end-devices. They cost several hundred euros with the cost of the SX1301-capable board alone to be more than a hundred euro. Here, again, the approach can be different in the context of agriculture/micro and small farm business: simpler "single connection" gateways can be built around an SX1272/76 radio module, much like an end-device would be. Therefore, rather than providing large-scale deployment support, IoT platforms in developing countries need to focus on easy integration of low-cost "off-the-shelves" components with simple, open programming libraries and templates for easy appropriation and customization by third-parties. By taking an adhoc approach, complex mechanisms, such as advanced radio channel access to overcome the limitations of the low-cost gateway, can even be integrated as long as they remain transparent to the final developers.

The WAZIUP LPWAN gateway will be built from off-the-shelves components such as the Linux Raspberry PI platform to benefit from the openness and the efficiency of a Unix-based system. This solution has high price/quality/reliability tradeoff as the cost of such gateway can be less than 45 euro, enabling deployment of a gateway per application.

In the proposed gateway architecture we clearly want to decouple the specific lower level radio bridge program (provided by WAZIUP) from the higher-level data post-processing stage that can be customized by third parties. The data post-processing stage will be developed in high-level language such as Python and a template will be provided for the developers. The main task of the post-processing stage is to push sensed data to the WAZIUP platform but it is possible to push data to any other third parties cloud/server/storage platforms as illustrated in Figure 44.

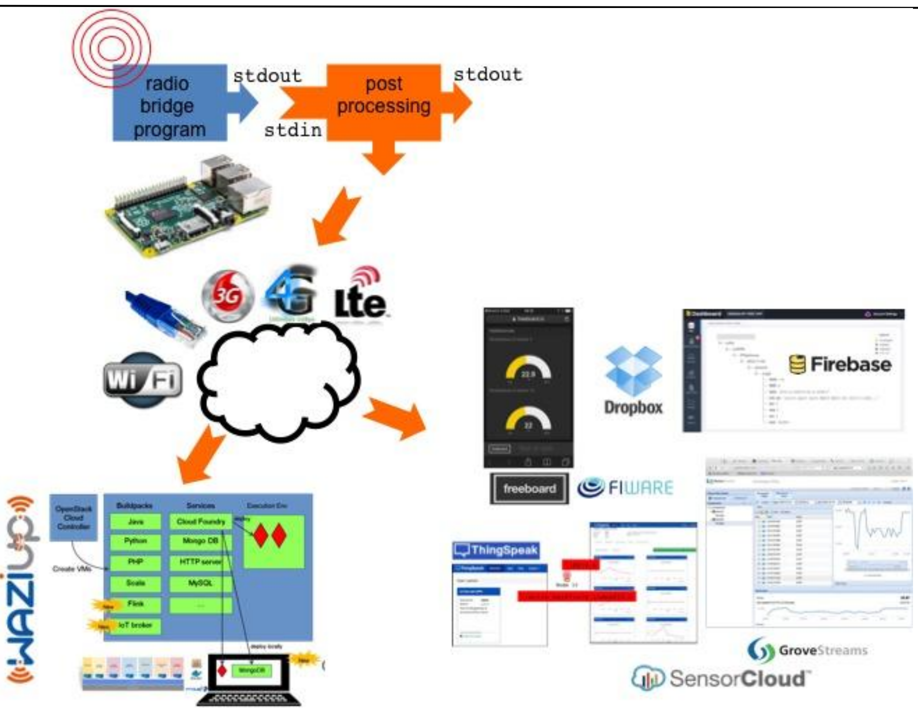


Figure 44: Data pushed to WAZIUP and third party platforms

The LPWAN gateway usually has Internet connectivity and will be connected to the WAZIUP platform using traditional (Internet) networking protocols. However, to handle cases where Internet connectivity is not available as explained in the Introduction, data are also locally stored on the gateway and can be accessed and viewed by using the gateway as an end computer by just attaching a keyboard and a display. The gateway can also interact with the end-users' computing device (smartphone, tablet) through WiFi or Bluetooth as depicted in figure. WiFi or Bluetooth dongles for Raspberry can be found at really low-cost (the Raspberry PI3 also comes with embedded WiFi and Bluetooth interfaces) and the smartphone can be used to display captured data and notify users of important events without the need of Internet access as this situation can clearly happen in very remote areas.

### 6.2.4 Network architecture

Most of public LPWAN architectures assume a Network server stage and an Application server stage. The Network server is usually a “public” entity for managing sensors and store sensed data. In an operator-based LPWAN infrastructure, the Network server will be deployed and owned by the operator. The Application server can be deployed and owned by the end-user. This is illustrated by the upper part of the Figure 45, taking the LoRaWAN architecture as an example.

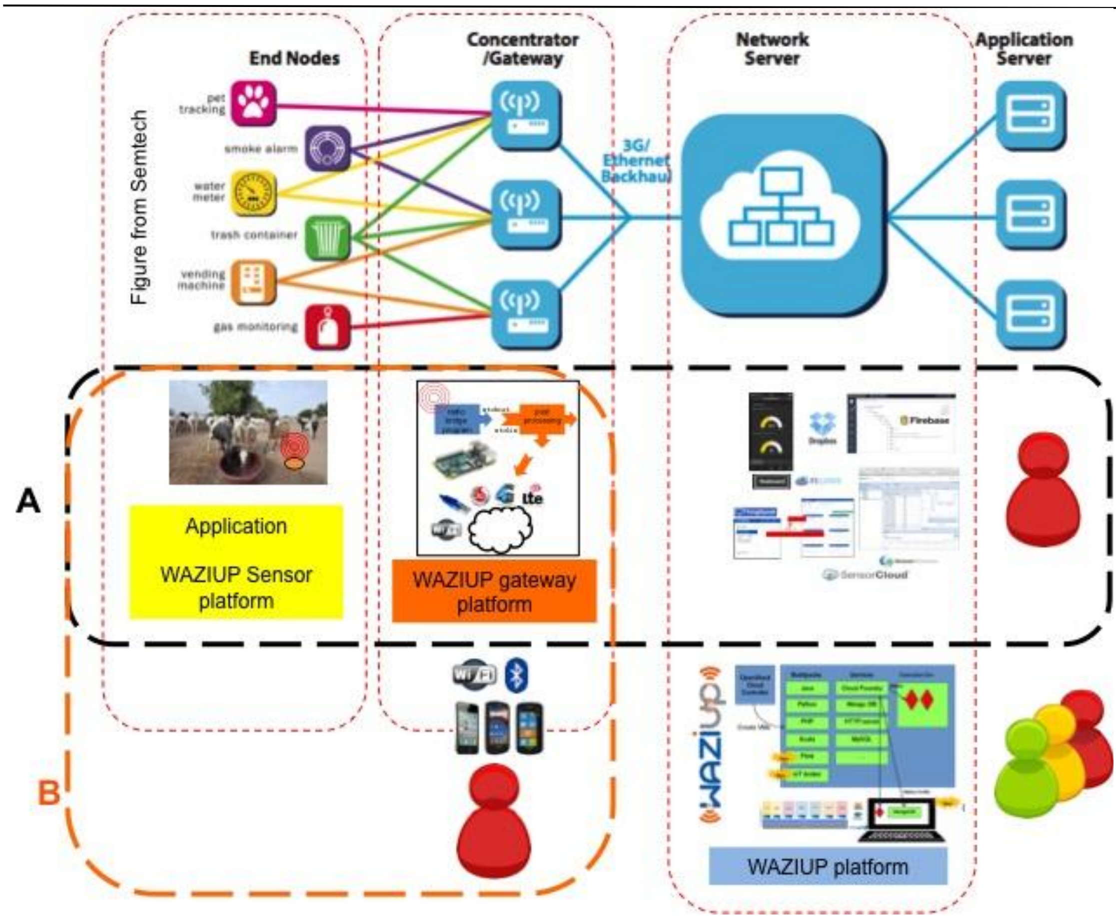


Figure 45: Network architecture

WAZIUP will provide the Sensor and Gateway platforms that can be deployed on a specific application-basis. Then, a subset of the WAZIUP platform can act as the Network server for advanced sensor and data management. However, for small-size, single-application scenarios and simple data collection profiles, this stage can be replaced by using public IoT data clouds (box A) or even local and direct interactions with the WAZIUP gateway platform (box B) without Internet connectivity requirements as it will be explained later on.

### 6.2.5 Proposed test beds

We will deploy a network test-bed at Gaston Berger University (UGB), Saint-Louis, Senegal, to validate the sensor and gateway platforms and to test various sensor settings in various rural environments. Computing facilities at UGB will host the WAZIUP platform to test advanced sensor and data management. The Internet access will also enable the small-size, single-application scenarios where public IoT data clouds will be used.

As can be seen in Figure 46, UGB has high buildings for the LPWAN antennas installation. By deploying LPWAN devices we can build a test-bed allowing LPWAN connectivity of IoT devices within a range of more than 15Kms in LOS in typical rural areas. In addition, the geographic location of UGB perfectly suit our needs as it is located within LPWAN radio range of the downtown Saint-Louis city as well as within range of many typical rural areas for test diversity, such as small villages, crop field and farms.

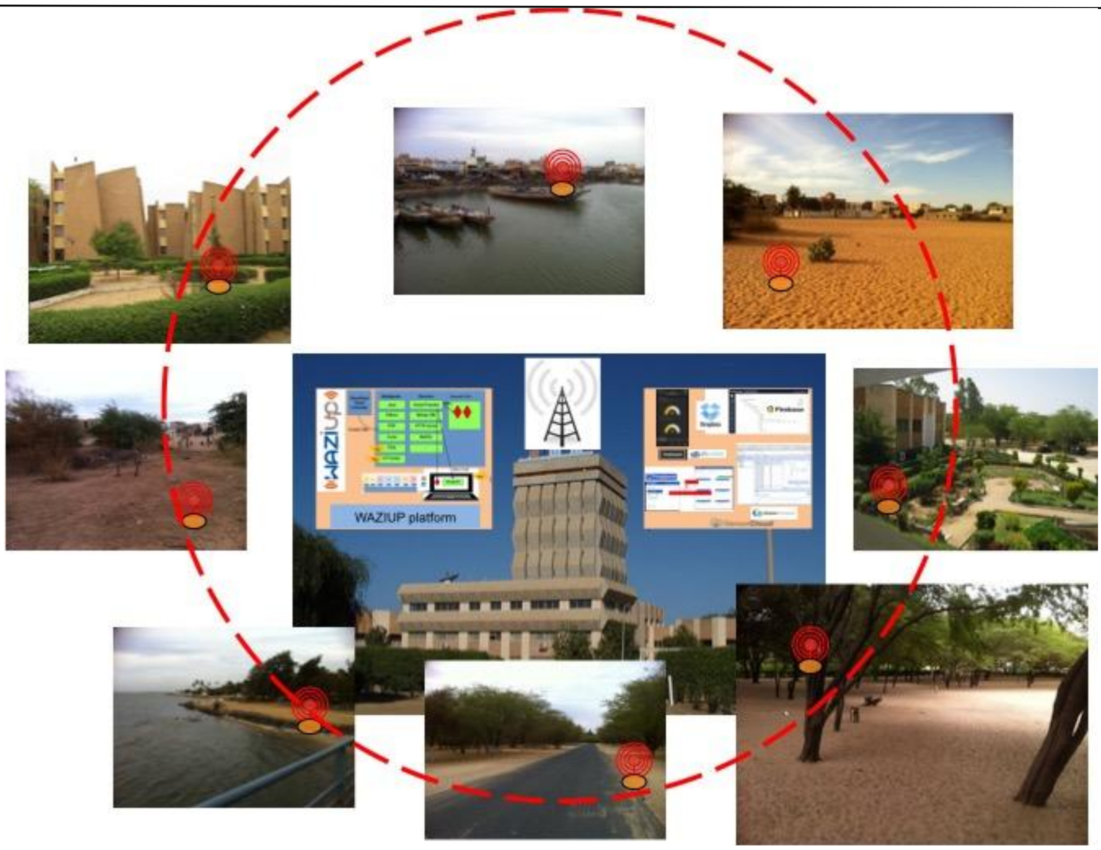


Figure 46: Deployment of sensor nodes around a gateway

### 6.2.6 Use case integration

Sensors are used to read parameters from the environment, which will be necessary by the use-cases, after that a network will be used to delivery into the WAZIUP platform the information collected, the next image shows the interaction between the components. The "Networking" box in the figure englobes both the gateway-sensors and the WAZIUP platform-gateway networking.

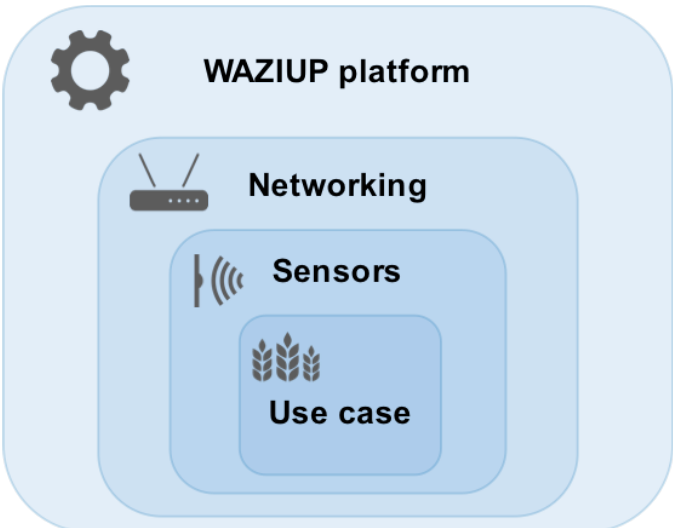


Figure 47: Use cases integration on WAZIUP platform

A use case describes a real world problem where information about the sensors used are also associated. There are several types of sensors each one with different characteristics, like autonomy, size, complexity, IO specifications among others, but the ultimate purpose of a sensor it's to read parameters from the physical world translating them into a signal ready to be used.

The information collected by the sensors must be transmitted to the platform in order to be used, and that must happen under different scenarios each one with different variants, which could be the number of sensors deployed, their locations, field conditions etc, this will determine the configuration of the network and which hardware to use.

An important feature provided by WAZIUP is the possibility to run the sensor-gateway system in an autonomous manner, without Internet connectivity nor access to dedicated servers. The gateway can therefore also store data locally and make them available through local computing facilities (e.g. laptop, smartphones, tablets) for standalone surveillance applications. Figure 48 illustrates the various scenarios that WAZIUP will support: (top) gateway will Internet connectivity provided by cellular technologies, (middle) gateway will Internet connectivity provided by a WiFi (possibly ADSL-based) access, (bottom) gateway without Internet connectivity, providing connectivity (short range) to local computing facilities (e.g. laptop, smartphones, tablets).

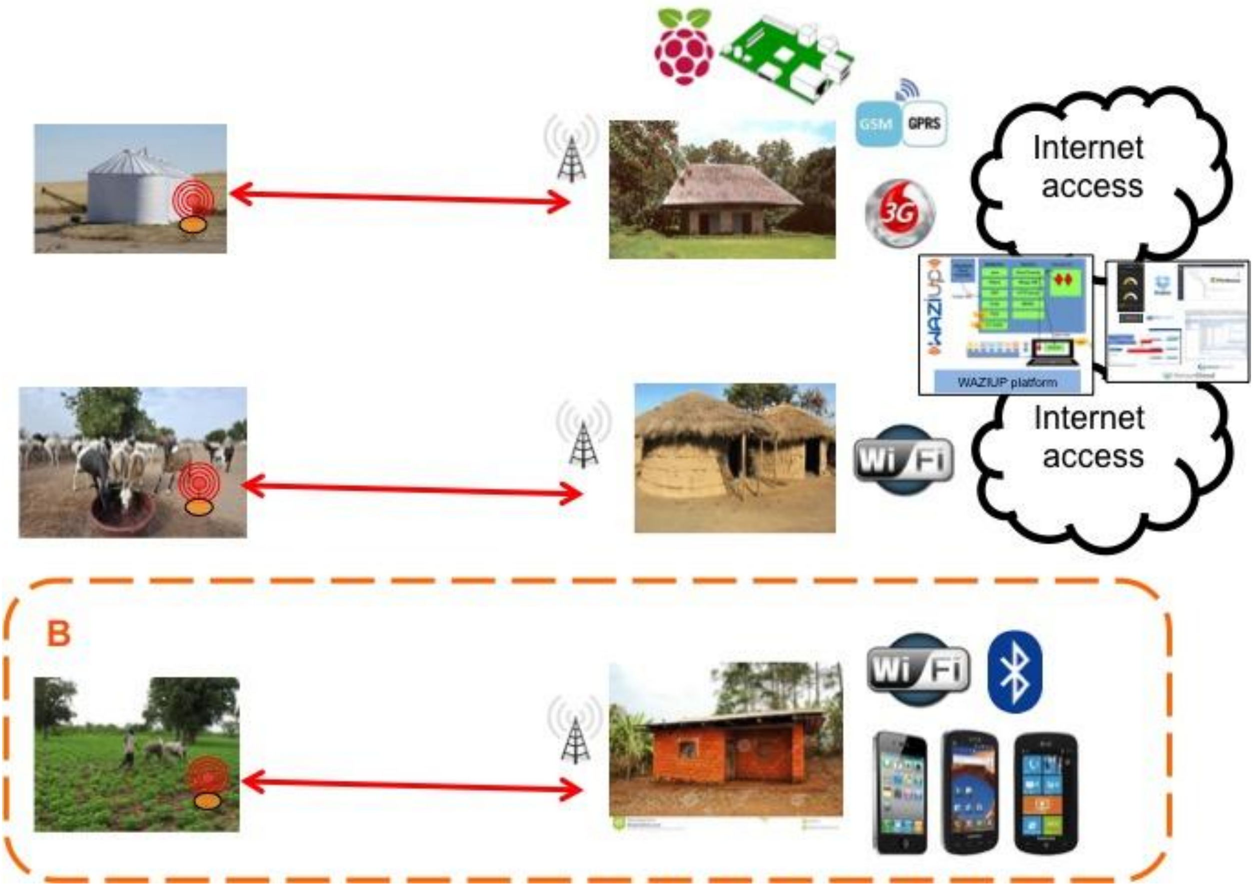


**Figure 48: WAZIUP deployment scenarios**

After the integration of the sensors and the gateway, the WAZIUP platform will be ready to receive and process information coming from the sensors.

## 6.3 Deployment and Installation

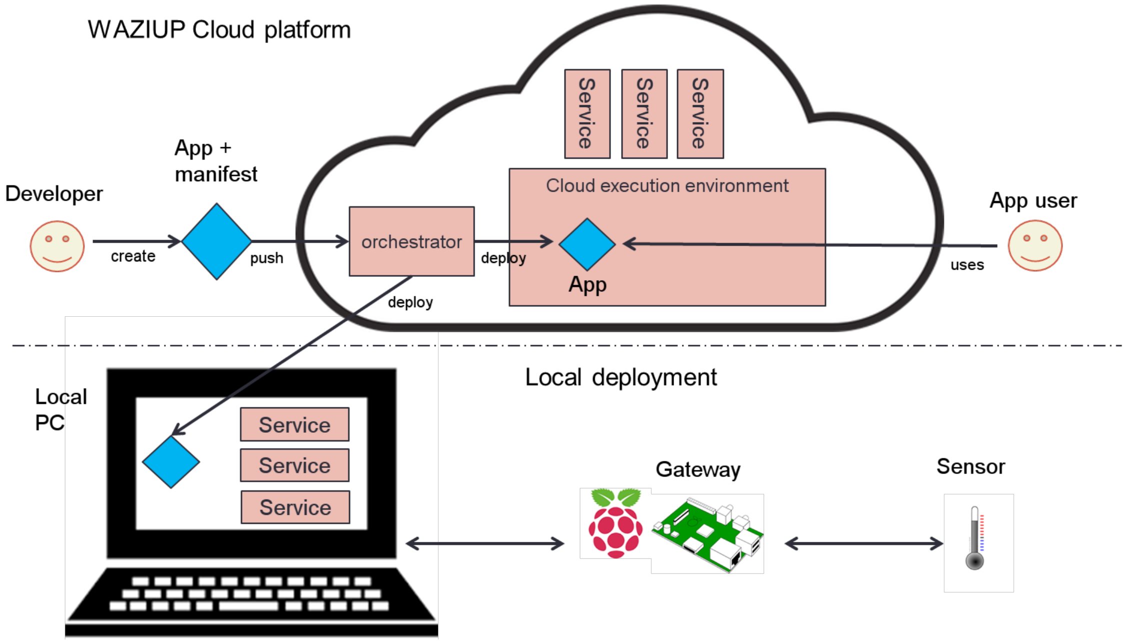


Figure 49: WAZIUP local and global deployments

WAZIUP will be deployed and accessed in an African context, where internet access is sometime scarce. Therefore, WAZIUP has a very strong constraint to be able to work with low internet connectivity. To fulfil this requirement, we propose a Cloud structure in two parts: the global Cloud and the local Cloud, as can be seen in Figure 49. The global Cloud corresponds to the Cloud in the traditional sense. The local Cloud corresponds to the gateway and an optional connected computer. The idea of WAZIUP is to extend the PaaS concept to the local Cloud. On the left hand side of the picture, the application is designed by the developer, together with the manifest file. It is pushed on the WAZIUP Cloud platform. The orchestrator then takes care of compiling and deploying the application in the various Cloud execution environments. Furthermore, the orchestrator drives the instantiation of the services in the Cloud, according to the manifest. The manifest is also describing which part of the application need to be installed locally, together with corresponding services. The local application can then connect to the gateway and collect data from the sensors.

# 7 DATA MODELS AND DATA FLOWS

This section introduces the data model and data flow of the WAZIUP system. The former refers to the representation of different entities and their respective attributes, as well as the relationships between them. Data flow depicts the flow of data generated by the different entities of the system. We first introduce some functional requirements, before discussing the proposed data model and data flows.

## 7.1 Functional Overview

There are difficult challenges to overcome when building an IoT platform. Most important challenges include : *data discovery, access, search, integration, interpretation and scalability*.

- Data discovery refers to the different processes in finding appropriate sensing resources and data sources.
- Data access make sensing resources and data open and available.
- Once data discovery and access are enabled, querying for sensor data is the last building block that is integrated in the *Discovery, Access* and *Search process*.
- Data integration refers to the ability to support different sensors' data when heterogeneous sensors are involved in the IoT application.
- Interpretation deals with the process of transforming sensors data into knowledge easily interpretable by different people.
- Scalability refers to the ability to deal with great amount of data and to maintain a good performance of the information (or knowledge) system in the context of data overload.

## 7.2 Data Model

The data model presented here is made of two parts: sensor data model and user-application data model. Firstly, the sensor data model describes the different entities in relation with sensors. User-application model defines the relationship between services and sensors data considering the different users and the applications of the platform. An intermediary section is introduced secondly, to highlight sensors data heterogeneity. The need of a unified data model is established based on the different applications that are likely to be connected to the platform.

### 7.2.1 Sensor Data Model

As can be seen in Figure 50, several entities are composing the sensor data model: *Sensor, Sensor data*, *Application*, *User*, *Configuration and finally Sensor register*. *Sensor* is a data structure holding the sensors meta-data, such as owner and position of the sensors. *Sensor data* is a data structure holding the data generated by the sensors themselves, such and temperature or humidity. The *Sensor register* is a data structure allowing recording which sensor is being used by which application. Sensor owners register sensors and each sensor is configured before its introduction in the platform. The system shall be able to store information of sensors configuration, the sensor owner and the different applications each sensor participates.

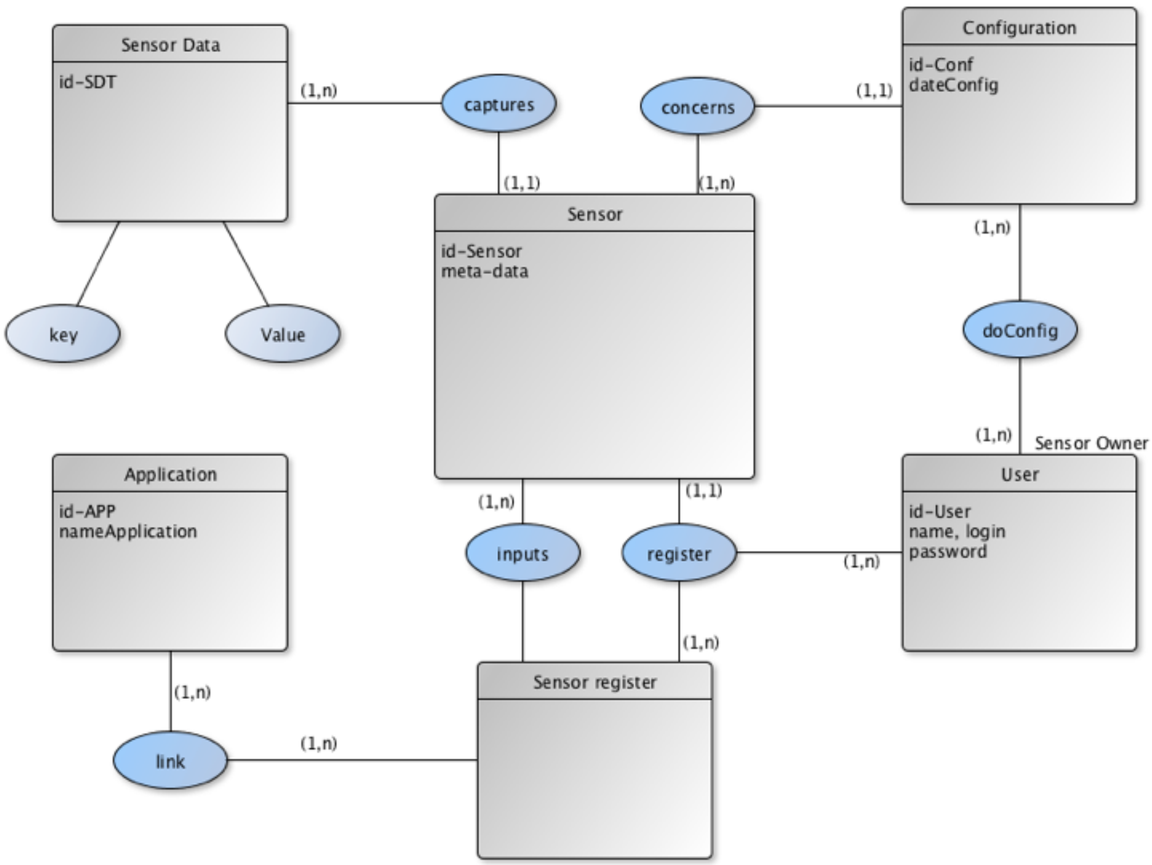


Figure 50: Sensor data model

### 7.2.2 Sensor Data Heterogeneity

The *WAZIUP* platform is an open platform intended to support diverse applications. Data generated from the application are likely to be in different formats. The diversity of the sensors that may be involved in applications leads to a wide variety of data formats. The figure above illustrates the data heterogeneity aspect yielded by sensors heterogeneity. This addresses the matter of data integration. This latter is to allow different data formats to communicate and ease data exchange. XML-based technologies and standard sensors modeling language such as SensorML and Semantic Sensor Web (SSW) are well suited for data representation, and can overcome the constraints of handling different data formats. Moreover, they enable semantic data manipulation, and their integration with ontologies and other semantic tools for sensors. The need to target unified data model is one step forward in data interpretation. This latter draws the objective to transform sensors data into knowledge easily interpretable by different actors.

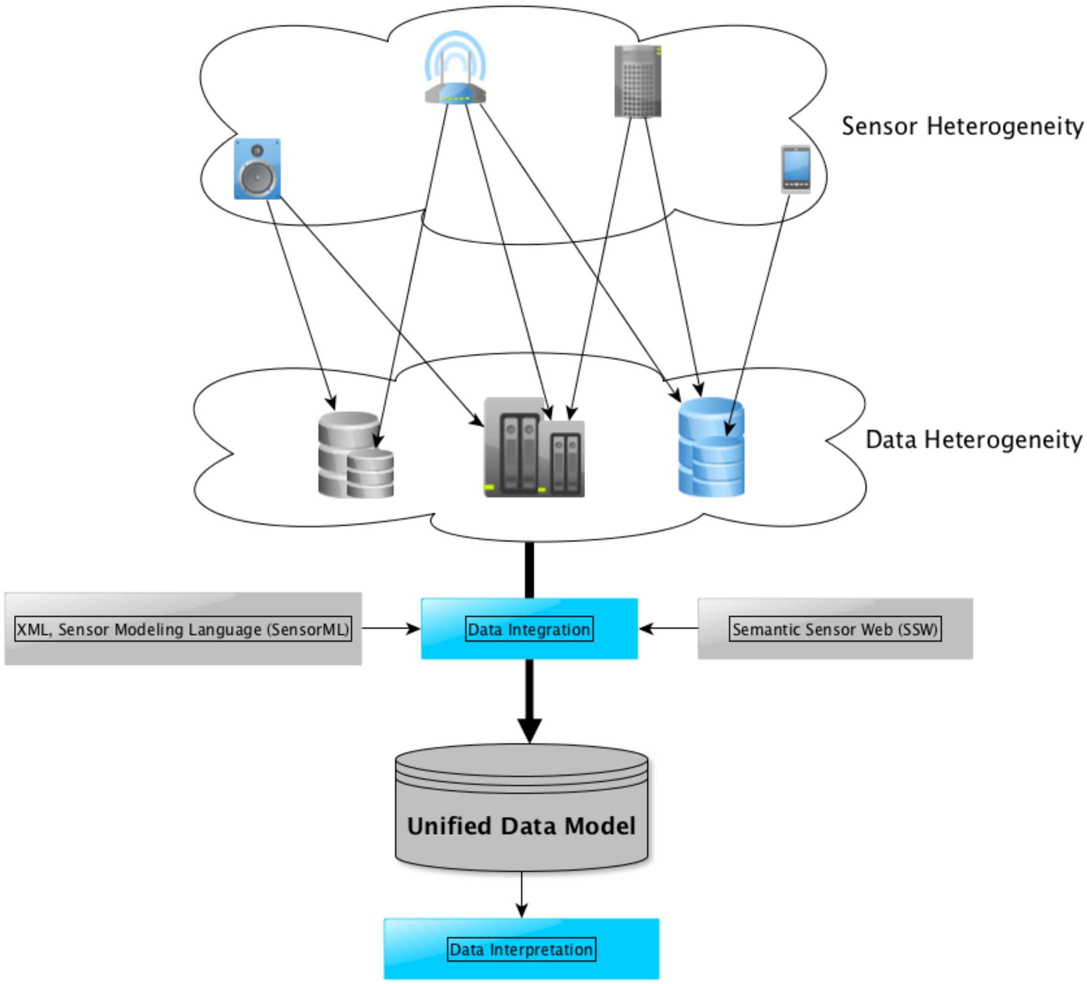


Figure 51: Sensors data heterogeneity

### 7.2.3 User-Application Data Model

As can be seen in Figure 52, The user-application model is concerned by two entities: *User* and *Application*. *Users* capture the different roles that are involved in the platform. These are: *Developer*, *App User* and *Data provider*. The *Developer* is responsible for application development and deployment. The *App User* role refers to the different users of an application. *Applications* are used either locally or remotely, and feedbacks are provided. *Data provider* entity links the model with service providers. *Services* are used by different applications, and applications may use different services. The system shall be able to track the service used by each application, as well as the corresponding providers.

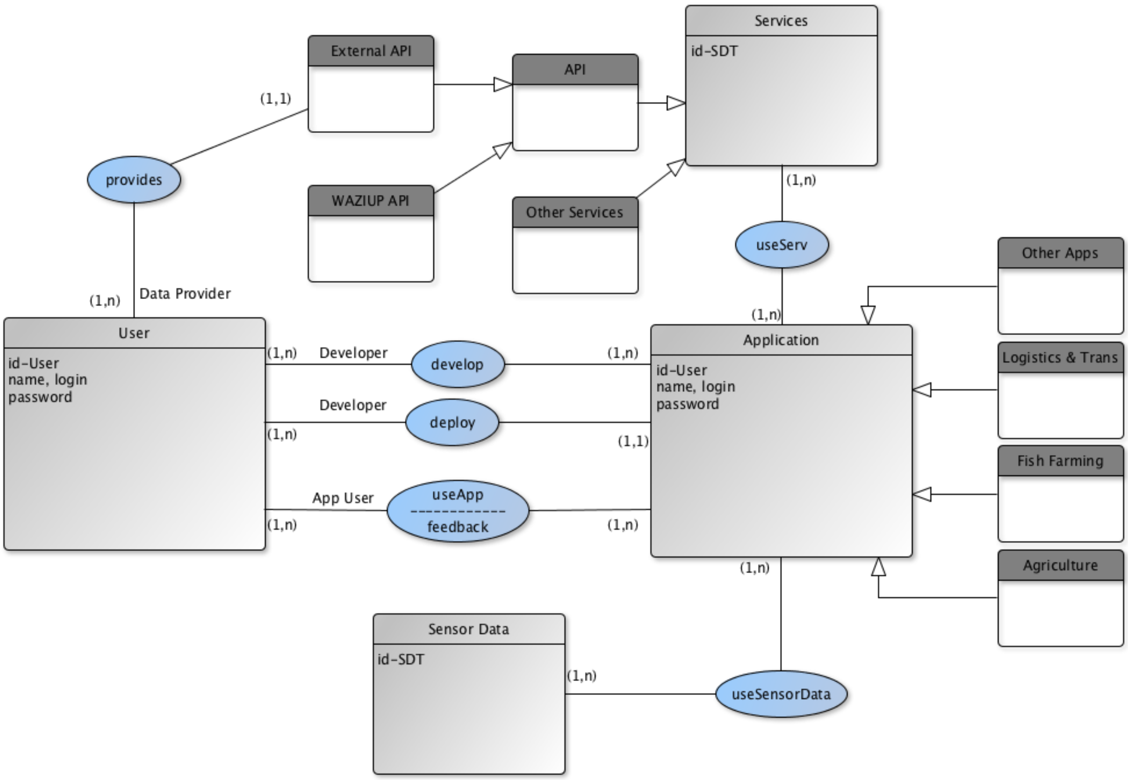


Figure 52: User-Application Data Model

## 7.3 Data Flow

The Figure 53 is a data flow model that pictures the different sources of data, and the different operations and entities that are associated with it. Legend is shown in the top right corner. From a data perspective, the data flow system can be divided into three parts: user's data, applications' data and sensors data. The first part relates to personal data, login data. Performed actions are "*login*", "*user management*" and "*registering*". The second part refers to applications data. The corresponding actions surround application development, deployment and management. The third one describes data generated from sensors data (sensor meta-data data, sensed data).

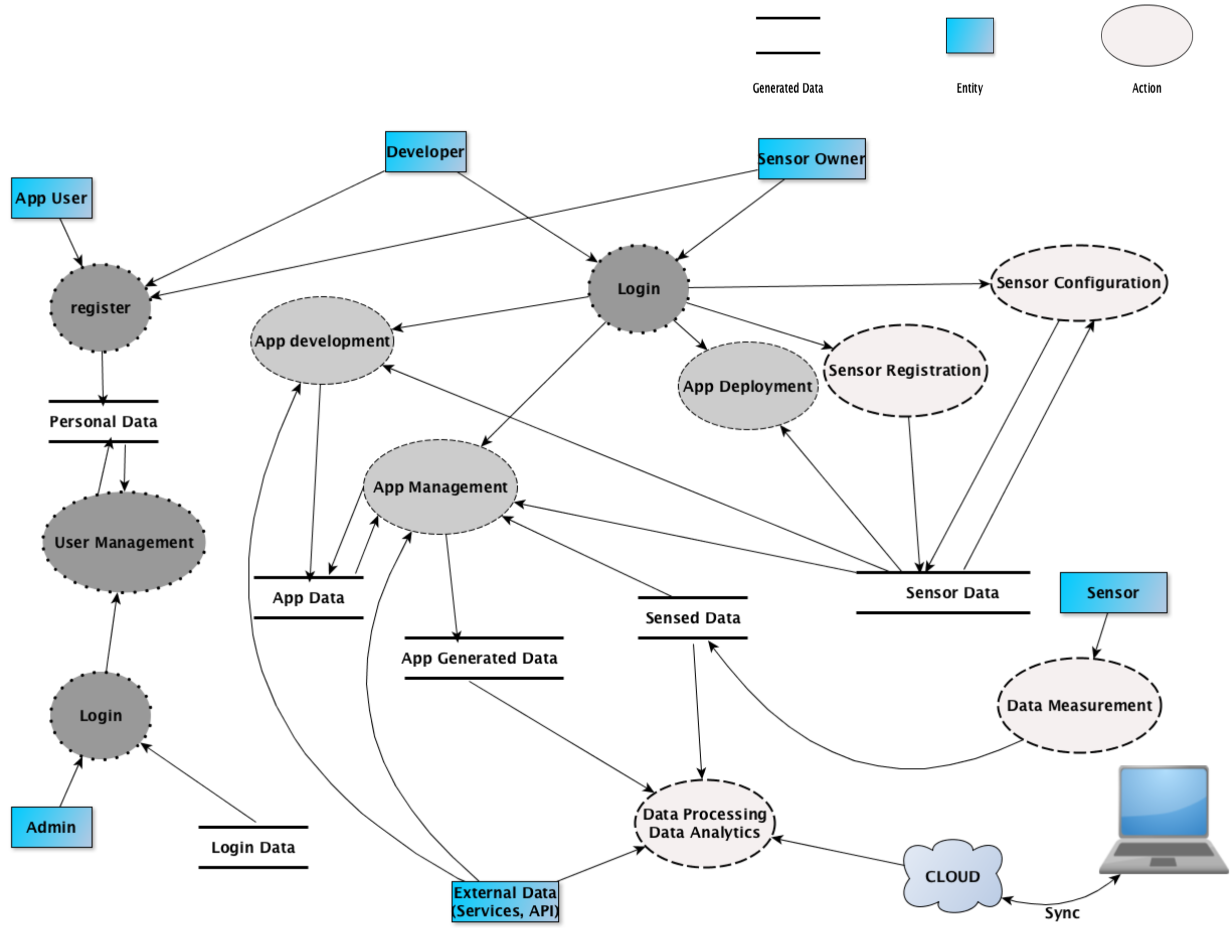


Figure 53: Model of data flow generation

## 7.4 Sensor database model

This model describes how is represented the data used on WAZIUP, that is retrieved through measurements where each one gives information about specific inputs from the environment, besides that, this section describes how the data is organized taking into account different domains.

The data described in this model represents several characteristics from the environment, that are retrieved with measurements obtained through readings of physical inputs taken by sensors, a measurement is composed by 1) a sensor, 2) the parameter and 3) the unit, allowing to know what is being measured and how it's being done. Figure 54 depicts possible compositions of measurements.

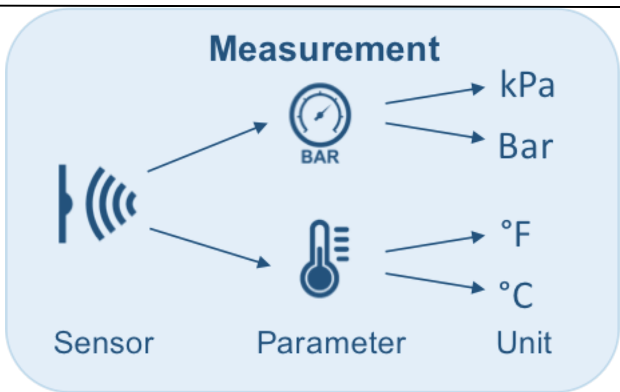


**Figure 54: Measurement representation**

A sensor collects data from the environment by receiving it from several physical inputs, then different measurements are produced taking into account the type of data read from the environment. In some cases, a single sensor can be used to collect data from multiple sources related with a specific domain, for example a weather sensor could be used to obtain data related with the temperature, humidity, wind speed, etc.

On a measurement, a unit gives meaning to the information that is read from the environment, then it is possible to know what is being read from the environment and how the data should be handled. In addition, relations can exist between units that can be compared being possible to convert the data between measurements represented by those units.


**Figure 55: Unit relation**

The Figure 55 illustrates a relation between the unit Volt (V) and Millivolt (mV), and with this relation is possible to obtain in Millivolt something measured in Volt. Relations of this type could be used on other units, facilitating the comparison of measurements with different orders of magnitude.

The “parameter” describes what is being read from the environment. It also includes the possible units that can be used coupled with the parameter. For example, ‘Water temperature’ is a parameter that is read from the water, and it could be represented using degree Celsius (°C), degree Fahrenheit (°F), etc. In some cases, a parameter can be represented by more than one unit, which allows for example to represent a measurement taking into account its order of magnitude (mV, V, etc) or different system of units (imperial, metric).

Besides the items that compose a measurement, each one can have its own properties which contain information about the characteristics and limitations of the data that is described by each measurement.

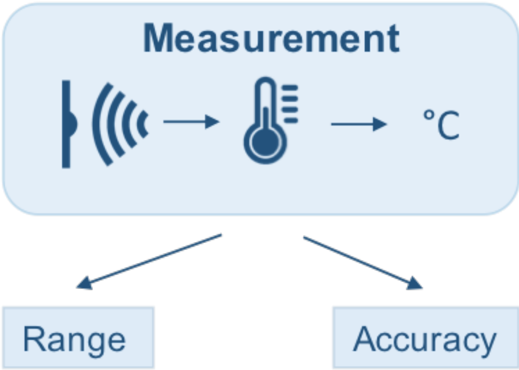


Figure 56: Measurement properties

Figure 56 shows the properties that can be associated to each of the collected measurements: Range and Accuracy.

- **Range:** Represents the maximum and minimum value that can be obtained from a measurement (eg, for dissolved oxygen 0 to 20 mg/L).
- **Accuracy:** Describes the maximum deviation between the read value on a measurement and the real value (eg, for temperature ±0.15 °C).

The domains are here used so to group the sensors together based on the environments and on the type of parameters they can read. Each domain has its own specific characteristic and need data regarding completely different parameters. The data gathered by one sensor can be useful in more than one domain, as depicted in Figure 57. The model represents the domains that are defined on WAZIUP, each one having a specific focus (e.g. Health, Agriculture, Fish farming, etc.).

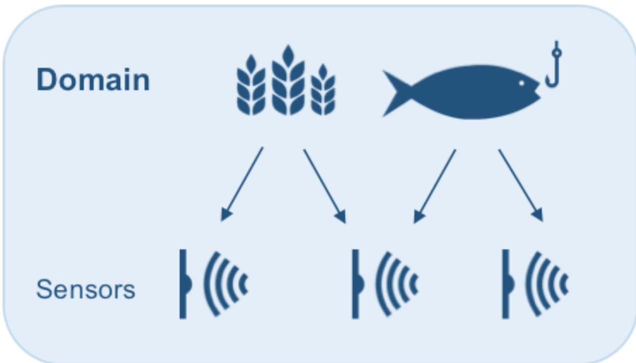


Figure 57: Domains representation

# 8 PRELIMINARY DESIGN

This section will define the design for each component. This includes technology choices. As WAZIUP is an open-source project, the technology chosen is open-source.

## 8.1 Application Platform components

The Orchestrator role is to read the manifest, trigger the compilation and deployment of the applications, instantiate the services needed by the applications and finally request the sensor and data sources connections. The orchestrator role is typically akin to a PaaS engine. The choice is constrained by the choice of the execution environment (probably Docker). Current candidates are Deis[5], Flynn[6], Tsuru[7] and Octohost[8]. Deis is an open source PaaS that can deploy and manage application. Deis builds upon Docker and CoreOS to provide a lightweight PaaS with a Heroku-inspired workflow. It is developed in Python and Go. Flynn is a PaaS manager based on Docker. It is developed in Go. Flynn is designed to run anything that can run on Linux, not just stateless web apps. Flynn includes built-in database appliances (just Postgres right now) and handles TCP traffic as well as HTTP and HTTPS. Tsuru is an extensible and open source Platform as a Service (PaaS) that can handle application deployment. Tsuru is an open source polyglot cloud application platform (PaaS). Tsuru is written in Go. As an application developer, you can write apps in the programming language of your choice, back apps with add-on resources such as SQL and NoSQL databases, including memcached, redis, manage apps using the tsuru command-line tool and finally deploy apps using the Git revision control system. Octohost is a simple web focused Docker based mini-PaaS server. Its focus is to deploy websites. As such, it doesn't seem to be enough for WAZIUP since we will also deploy non web-based applications.

The technology selected for the Execution Environment is the concept of container[9]. They are light-weight execution environments. They are also portable and start very quickly, as a difference with virtual machines. There are two candidates for the container management system: Docker[10] and Kubernetes[11]. Both Kubernetes and Docker are "Containers as a Service" (CaaS). They offer containers hosting and management. Docker seems the best candidate though, because of the wide community and activity around it. Furthermore, the EU project AGILE[12] is planning to enable the use of Docker in the IoT gateway, which is a very important point for WAZIUP.

The selected technology for Rapid Application Development is Node-RED. This tool developed by Google allows designing a data flow application graphically. Node-RED then produces a NodeJS application.

The UI manager should support the following technologies: SMS/USSD, automated voice calls and web interfaces. The technologies selected are routeSMS[13] for the SMS/USSD, freeSwitch[14] for the voice calls and standard technologies such as Bootstrap, JQuery, PHP for the web interfaces.

[5] http://deis.io/
[6] https://flynn.io/
[7] https://tsuru.io/
[8] https://octohost.io/
[9] https://linuxcontainers.org/fr/
[10] https://www.docker.com/
[11] http://kubernetes.io/
[12] http://www.agile-project-iot.eu/
[13] www.routemobile.com

## 8.2 IoT Platform components

The IoT bridge role is to communicate with the devices in their own protocols. We propose the FIWARE component IoT agent that can be used or extended to this aim and we can also develop our own IoT bridge depending on the device supported communication protocol. Candidates for the sensor registry component are:

- Orion[15] IoT broker that has an internal component serving as a sensor registry,
- NEC ConfMan[16] designed to work with the IoT broker generic enabler,
- LwM2M device management protocol which also provides a sensor registry.

## 8.3 Stream and data analytic

Regarding the data broker component, Orion and Kafka has been selected.

For the Storage manager component, some specific requirements have to be considered. Due to the diversity of applications and the corresponding data format, approaches usually adopted in classical database system can be heavy, or may lead to a lack of flexibility. The WAZIUP platform is an open platform, dedicated to target different types of applications. The non-structuration of data combined to the diversity in the upcoming application, may raise the possibility to handle data based upon specific requirements. These requirements are specified on the requirements CSV file. They can be summarized into the following points below.

- Data replication between local and cloud.
- Data synchronization between local and cloud.
- Data persistence guarantees accessibility in case of connection failure.
- The system must support scalability by allowing data availability and distribution.
- Fault tolerance guarantees that every request must be fulfilled independently of the status of the addressed server.
- Handling of different data types implies choosing a unique data format in order to ease information exchange.

Candidates for the data storage are Apache HBase[17], Apache Cassandra[18] and MongoDB[19].

The WAZIUP platform targets a low Internet connectivity constraint. Therefore, a two-part cloud structure is adopted: a global cloud corresponding to the traditional cloud, and a local cloud involving a gateway and an optional connected computer. The orchestrator is responsible for managing applications compiling and deployment in the local and global cloud execution platform, and leads the instantiation of the services according to the application's manifest. Services instantiated by the orchestrator shall be stored locally, to allow the application run without the need to connect to the global cloud.  The selected data analytic tools are Apache Flink [20]and Apache Spark[21] for the stream processing (see Section 9).

[14] https://freeswitch.org/
[15] http://catalogue.fiware.org/enablers/publishsubscribe-context-broker-orion-context-broker
[16] https://github.com/Aeronbroker/NEConfMan
[17] https://hbase.apache.org
[18] http://cassandra.apache.org
[19] https://www.mongodb.com
[20] https://flink.apache.org
[21] https://spark.apache.org

## 8.4 Security and privacy

The candidates for both the Identity Manager and Authorization Manager are Gluu[22] and OpenAM[23]. The Gluu Server is an identity and access management suite comprised of free open source software (FOSS) components, and supports the following open web standards: OpenID Connect, SAML, UMA, SCIM, FIDO U2F, and even LDAP. OpenAM is an “all-in-one” access management solution that includes Authentication, SSO, Authorization, Federation, Entitlements, Adaptive Authentication, Strong Authentication, and Web Services Security.

The candidate for the Privacy Manager is ARX[24]. This tool is a powerful de-identifier allowing to anonymize the data.

[22] https://www.gluu.org/
[23] https://forgerock.org/openam/
[24] http://arx.deidentifier.org/

# 9 STATE OF THE ART

This section gives a quick overview on the IoT and big data platforms state-of-the-art. It will present the IoT architecture and the IoT standardization efforts and a study of the most-used open source tools for embedded Database Management System (DBMS) and big data processing.

## 9.1 IoT platform

The European founded Alliance for IoT Innovation[25] (AIOTI) launched in 2015 is now the largest European IoT ecosystem. It builds upon several working groups, one of them (WG3) being focused on IoT High Level Architecture (HLA) to derive generic architecture for IoT. The next two next sections describes the IoT domain model and the IoT functional level as defined by the AIOTI group WG3.

### 9.1.1 IoT domain model

The proposed domain model builds upon the results of the IoT-A project which concludes that a user should not interact directly with physical things but get this interaction mediated by an IoT service exposing a virtual representation of the thing and interacting with the thing through an IoT device (see Figure 58).

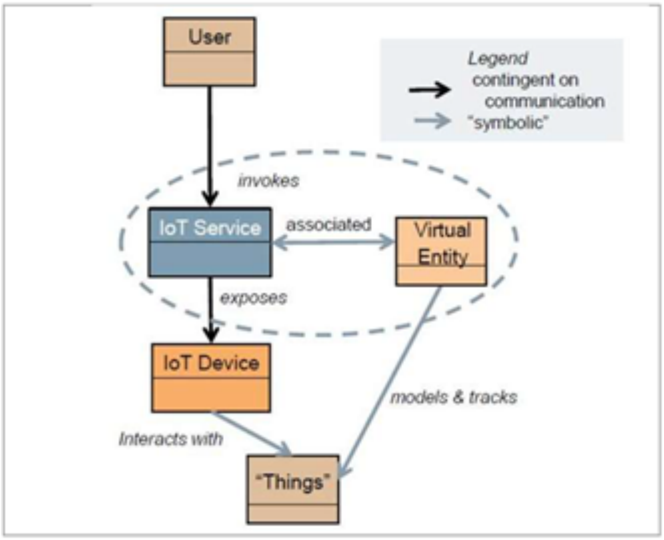


Figure 58: AIOTI WG3 domain model

### 9.1.2 IoT functional model

The AIOTI functional architecture[26] is depicted in Figure 59.

[25] http://www.aioti.eu

[26] High Level Architecture (HLA) Release 2.0, IOTI WG03 – IoT Standardisation 2015

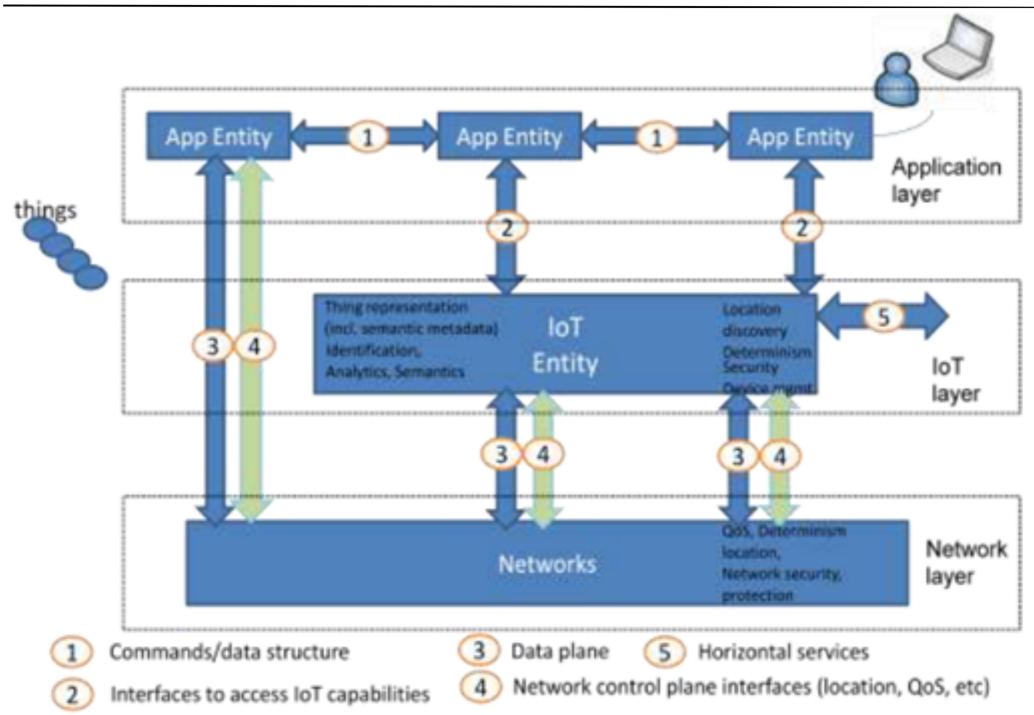


**Figure 59: AIOTI HLA functional model**

The AIOTI functional model is based on 3 layers: applications layer, IoT layer and the network layer. The following functions are defined:

- *App Entity:* is an entity in the application layer that implements IoT application logic. An App Entity can reside in devices, gateways or servers. A centralized approach shall not be assumed. Examples of App Entities include a fleet tracking application entity, a remote blood sugar monitoring application entity, etc.
- *IoT Entity:* is an entity in the IoT layer that exposes IoT functions to App Entities via interface 2 or to other IoT entities via interface 5. Typical examples of IoT functions include: data storage, data sharing, subscription and notification, firmware upgrade of a device, access right management, location, analytics, semantic discovery etc. An IoT Entity makes use of the underlying Networks' data plane interfaces to send or receive data via interface 3. Additionally, interface 4 could be used to access control plane network services such as location or device triggering.
- *Networks:* may be realized via different network technologies (PAN, LAN, WAN, etc.) and consists of different interconnected administrative network domains. The Internet Protocol typically provides interconnections between heterogeneous networks. Depending on the App Entities needs, the network may offer best effort data forwarding or a premium service with QoS guarantees including deterministic guarantees.

The AIOTI HLA enables the digital representation of physical things in the IoT Entities. Such representations typically support discovery of things by App Entities and enable related services such as actuation or measurements. The representation of things typically contains data, such as measurements, as well as metadata. The metadata provide semantic descriptions of the things in line with the domain model and may be enhanced/extended with knowledge from specific vertical domains. The representation of the things in the IoT Entities is typically provided by App Entities or IoT Entities residing in devices, gateways or servers. A one to one mapping between a physical thing and its representation shall not be assumed as there could be multiple representations depending on the user needs. Figure 60 provides the relationships between the physical things, their representations and the link to semantic metadata which are an instantiation of the domain model described earlier in this document.

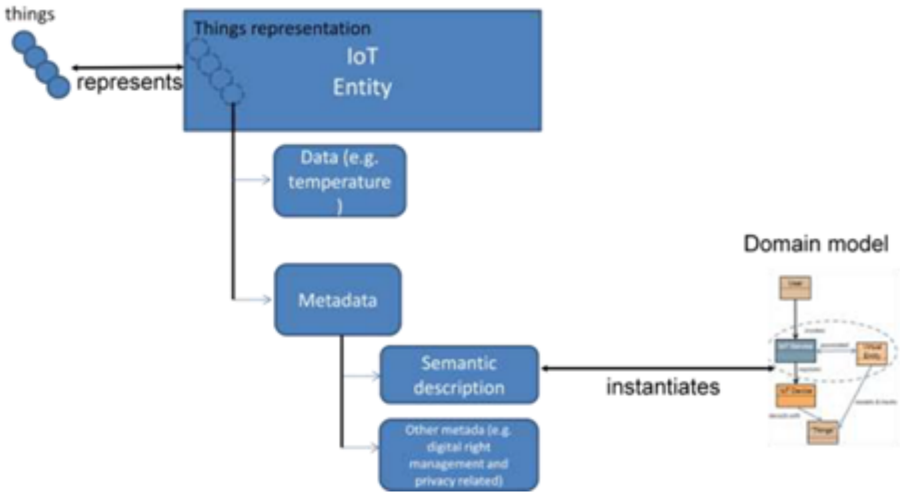


Figure 60: Relationship between a thing, a thing representation and the domain model

### 9.1.3 IoT Standards

Standardization landscape for IoT is constantly evolving and every week new standards and new industrial alliances are announced. This effervescence created by the increasing industrial interest on IoT also comes from the need of industries to find their place within the business ecosystem perturbations brought by IoT. For many years, starting with the IoT-A project in Europe, attempts to identify the winning architecture have emerged but none did result in being globally accepted This highlighted the fact that IoT will build on interworking of heterogeneous solutions and that interoperability will be a key success factors for IoT.

The AIOTI WG3 is focusing on standardization. This working group, gathering attraction from global IoT industry players as well as standard organizations and Alliance initiated a mapping of standards and specification for IoT. In addition, it proposed an HLA building upon all past attempts to derive generic architecture for IoT. This work is now being reworked and expanded within a Specialists Task Force (STF) launched by ETSI (STF505) to analyze gaps existing between the current standards landscape and the needs of use cases.

WAZIUP intends to deploy IoT and big data solutions in Africa and to offer the possibility to local communities to make use of these solutions for their own needs. For that reason, openness of proposed solution is critical for its success. Openness has to be understood as:

- *Solutions based on open standards:* standards and specifications used within the solutions have to be published, freely accessible and usable by anybody. In addition, the contribution process to this specification should preferably be open.
- *Solutions having open-source implementations:* WAZIUP is a deployment project and as such, needs to access implementations of selected standards. These implementations have to be proposed as open-source with licensing schemes compatible with intended WAZIUP use cases.

For these reasons, WAZIUP will build upon on-going EU initiatives, starting with HLA, based on ISO/IEC/IEEE 42010 standard, under discussion with AIOTI WG3.

## 9.2 Big data platform

This section compares some open source big data and stream processing tools. We first introduce a comparison of available open source embedded Database Management System (DBMS), and then present tools for big data processing.

### 9.2.1 Local Storage Technologies

According to the WAZIUP platform requirements, different criteria have to be considered for a technological choice. Table 77 depicts the available technological choices and the relevant aspects, which respond to these specific requirements.

Table 77: Comparison of available technological choices

| Name | Size | Real-time | Open Source | Concurrency | Language Extensions | Data structure | Scalability |
|---|---|---|---|---|---|---|---|
| Perst | Not Defined | Not defined | Yes | No | Java | Object-Oriented | Yes |
| Berkeley DB (BDB) | 256 Tb | Yes | yes | yes | Java, C | Key-Value | Yes |
| SQLite | Not Defined | Yes | Yes | Yes | Java, C, Python | Relational | Yes |

### 9.2.2 Databases and data warehouses

**HDFS**, developed by Apache, is a distributed, scalable and portable file-system written in Java for the Hadoop Framework. It has been designed for large dataset analysis and by its structure has high fault tolerance. It is the basis upon which everything works in the Hadoop Ecosystem.

Built on top of HDFS, Apache **HBase** is a distributed, non-relational column oriented datastore. HBase is designed to efficiently address random access and fast record lookup. It has the capability to handle extremely large tables of data with low latency. Though, this data storage tool should be used when random and real-time read/write access to data is needed and when many thousands of operation per seconds need to be performed on large datasets (up to petabytes).

Apache **Hive** is a data warehouse infrastructure that can manage and query unstructured data as if it were structured. As a full component of Hadoop Ecosystem, it uses MapReduce for execution and HDFS for storage. It has its own SQL-like language (HiveQL) that brings expressiveness to the queries. This storage mode should be used for SQL-like queries and when higher level language than MapReduce is needed.

Used by big companies who can't afford to lose data (Apple, Netflix, Spotify …), Apache **Cassandra** is a column oriented database of structured data. The data are highly available through column indexes and are automatically replicated through multiple nodes for fault tolerance. Cassandra has a unique masterless "ring" design that is easy to setup and to maintain. This tool should be used when losing data is not the critical point and not affordable.

**MongoDB** allows to get rid of traditional table-based relational database. It is a NoSQL, relational, document oriented database. Documents are shared in JSON format with dynamic schemas (called BSON) which makes data integration sometimes easier. This database is very useful when data are consumed by multiple applications as many connectors have been developed.

### 9.2.3 Data publication and subscription

Apache **Flume** is a distributed, reliable and available service for efficiently collecting, aggregating and moving large amounts of streamed event data. Flume should be used if data are designed for Hadoop as it can move them to HDFS. It has many built-in sources and sinks and can process data in-flight using interceptors, which is useful for data masking or filtering. It is composed of agents and data collectors (and interceptors if needed).

For more general purposes, Apache **Kafka** is a high-throughput, distributed, publish-subscribe messaging system. It can replicate events, has low latency and is capable of data partitioning. Kafka is also easily scalable and this tool is very useful when data are to be consumed by multiple applications. It is composed of producer, consumers and topics.

Actually, we might not have to choose between Kafka and Flume, as both can work quite well together. If the workflow design requires streaming data from Kafka to Hadoop, using a Flume agent with Kafka source to read the data makes sense. This association is quite common and is called **Flafka**.

If a Data/Context Scenario is developed, we may need to use a context broker. **Orion Context Broker** (FIWARE platform) is a publish/subscribe platform that is able to register context elements and manage them through updates and queries. It is possible to subscribe to context information when some conditions occurs (e.g. an interval of time passed or the context elements have changed). Orion is a C++ implementation of the NGSI9/10 REST API binding developed as a part of the FIWARE platform.

### 9.2.4 Data processing

Choosing a data processing tool mostly depends on the outcome expected from the data. The most prominent tool for big data analysis is Hadoop MapReduce. Other tools exist and some are becoming more and more powerful.

MapReduce programming model contributed to the amazing progress of Big Data processing this past decade. By breaking down the work and recombining it in series of parallelizable operations, it is simple but very efficient and scalable to ten thousands of machines. It can run on inexpensive hardware, lowering the cost of a computing cluster. The latest version of MapReduce is **YARN**, called also MapReduce 2.0.

If a higher level of programming on top of MapReduce is needed, Apache **Pig** can be used. Pig has its own language (PigLatin) similar to SQL and works on top of MapReduce. Pig Engine parses, optimizes and automatically executes PigLatin scripts as a series of MapReduce jobs on a Hadoop cluster. It is easy to learn and opens Hadoop to data professionals who may not be software engineers.

First designed to work with HDFS on top of YARN, Apache **Spark** is a different system for processing data and can work out of Hadoop ecosystem with other data managements systems. It does not work with MapReduce and it can be up to a hundred times faster than MapReduce with its capacity to work in-memory, allowing keeping large working datasets in-memory between jobs, considerably reducing the latency. It allows a wide range of applications: batch and stream processing (micro-batch processing with 0.5s latency), machine learning (MLib), SQL (with Hive), graph Analytics (graphX). Supported language are Java, Python and Scala.

Demand for streaming analysis becoming more and more important in Big Data analysis, Apache **Flink** has been recently developed and is growing very fast. Flink is a streaming dataflow engine that provides data distribution, communication and fault tolerance. It has almost no latency as the data are streamed in real-time (row by row). It runs on YARN and works with its own extended version of MapReduce. Supported language are Java and Scala.

### 9.2.5 Machine learning

Machine learning is the union between statistics and artificial intelligence. It blends AI heuristics with advanced statistical analysis. We let the machine learn about the data, make decisions, and then apply statistics. Algorithms used for these tasks can be grouped in 3 domains of actions: classification, association and clustering. To choose an algorithm, different parameters must be considered: scalability, robustness, transparency and proportionality. Overlearning (or overfitting) of the model must be carefully checked.

Without any math or programming requirement, **KNIME** is an analytic platform that allows users to process the data in a user-friendly graphical interface. It allows training models and evaluating different machine learning algorithms.

If the workflow is already deployed on Hadoop, a machine learning library exists and is called **Mahout**. Spark also has his own machine learning library called **MLib**.

**H20** is a software dedicated to machine-learning, which can be deployed on Hadoop and Spark (support of Flink is in development). It has an easy to use Web interface, which makes it possible to combine big data analytics with machine learning algorithm in order to train models.

### 9.2.6 Data visualization and exploration

To visualize the data in real time, **Freeboard** provides a simple, real-time dashboard, commonly used in IoT world. There is a direct Orion Fiware connector for Freeboard. To connect with streaming engines, a JSON connector can be used. Design is simple and customization is not possible, but it is a very good dashboard to easily visualize raw data coming from sensors, before data analysis.

**Tableau Public** offers a good visualization and exploration tool on batch data. Tableau is a software where you can upload your data (previously extracted in .csv format). The visualisation tool is very powerful and allow a deep exploration the data. The open Source version of Tableau does not offer the data streaming capabilities (e.g. Kafka connectors). That makes Tableau Public a highly customisable, user-friendly and intuitive exploration tool for data that have already been processed and analysed.

To visualise data in real-time, after analysis (filtering, aggregating, correlating …), one of the best tool is probably **Kibana**. It is the visualisation tool coming with ElasticSearch. Elasticsearch is a search server based on Apache Lucene that provides a distributed, multitenant-capable full-text search engine with an HTTP web interface and schema-free JSON documents. It is really designed for real-time analytics, most commonly used with Flink or Spark Streaming.

# 10 CONCLUSION

With ICT technologies, Africa can dramatically improve its agricultural productivity by enabling the rapid and cost-effective deployment of advanced and real-time monitoring. However, deploying an IoT platform for Africa comes with many challenges. Among them, the most important are supporting low cost, low power, low bandwidth, and intermittent from Internet. Furthermore, widely accessible technologies such as SMS and voice calls need to be supported to reach the maximum users. In this part, we proposed an architecture for the WAZIUP IoT Big Data platform. The concepts that underpin the WAZIUP platform are three: the PaaS approach to IoT, the data processing capacity inspired from Big Data techniques and finally the local and global Cloud. The idea of extending the PaaS approach to IoT is to propose a platform dedicated to IoT developers that can reduce the time-to-market for an application by cutting the development costs. The Big Data techniques enable the processing of the huge amount of data produced by sensors. Those techniques allow creating actionable information and knowledge out of the raw data. Finally the local and global Clouds address the intermittent connection challenge: when Internet is not available, the user can still access some IoT functionalities from the local Cloud. This document detailed the architecture, requirements, interfaces, data models, hardware, deployment, and preliminary design of such a platform. It also gives a state of art of the useful technologies.

# 11 PROJECT CO-ORDINATOR CONTACT

Dr. Abdur Rahim

CREATE-NET

Via alla Cascata 56/D

Povo- 38123 Trento, Italy

Tel: (+39) 0461 408400

Fax: (+39) 0461421157

Email: abdur.rahim@create-net.org

# 12 ACKNOWLEDGEMENT

This document has been produced in the context of the H2020 WAZIUP project. The WAZIUP project consortium would like to acknowledge that the research leading to these results has received funding from the European Union's H2020 Research and Innovation Programme under the Grant Agreement H2020-ICT-687670.